

**USING OBJECT DESCRIPTIONS
IN A SCHEMA NETWORK
FOR MACHINE VISION**

Terry E. Weymouth

COINS Technical Report 86-24

May 1986

This research was supported by the Air Force Office of Scientific Research under grant F49620-83-C-0099, by the Defense Advanced Research Agency under contract DACA76-85-C-0088, and by the National Science Foundation under grant DCR-8318776.

**USING OBJECT DESCRIPTIONS IN A SCHEMA NETWORK
FOR MACHINE VISION**

A Dissertation Presented

By

TERRY EDWARD WEYMOUTH

**Submitted to the Graduate School of the
University of Massachusetts in partial fulfillment
of the requirements for the degree of**

DOCTOR OF PHILOSOPHY

May 1986

Department of Computer and Information Sciences

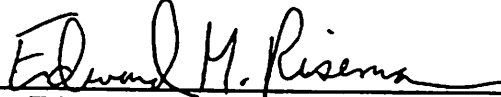
Terry Edward Weymouth
© 1986
All Rights Reserved

**Research supported in part by:
The National Science Foundation
DCR-8318-776,
Air Force Office of Scientific Research
F49620-83-C-0099,
and
Defense Advanced Research Projects Agency
DACA76-85-C-0008.**

Using Object Descriptions in a Schema Network
For Machine Vision

A Dissertation Presented
By
Terry Edward Weymouth

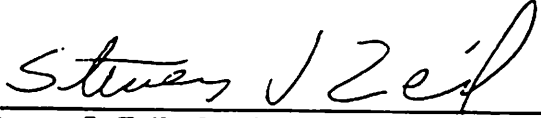
Approved as to style and content by:


Dr. Edward Riseman, Committee Chair


Dr. Allen Hanson, Member


Dr. Edwina Rissland, Member


Dr. Keith Rayner, Outside Member


Dr. Steven J. Zeil, Graduate Program Director
Department of Computer and Information Sciences

To
My Families

ACKNOWLEDGEMENTS

I start with a word of thanks to the members of my committee and to other faculty members whose advice I sought. Edward Riseman inspired and guided this research. He also did much to improve my use of language and style of writing (although you, dear reader, will see only those foibles which I managed to slip past him). The best of this document stems, in no small part, from his insistence upon "doing it right." Most of all, however, he was an unfailing supply of intellectual energy. My thanks, also, to Allen Hanson, who was consistently able to provide clarification and direction. When Ed was the force, Al was the pivot. In addition, Edwina Rissland and Keith Rayner served on my committee and were most helpful. Occasional guidance was also offered (but too much ignored) by Michael Arbib and Victor Lesser. My thanks to them.

The work described herein would not have been possible without the contributions and assistance of many other researchers. Principal among these are the staff, members, and leaders of the VISIONS research group at the University of Massachusetts. From my association with them, I inherited a rich culture of attitudes and concepts that guided my research. In addition, the implementation and testing of this research would have been impossible without the laboratory environment provided for and by the group. I am especially grateful for the efforts of that group to build and maintain a software base for computer vision research, without which this dissertation would not be.

Several fellow students contributed to the growth and flow of most of the ideas developed in this dissertation. Tom Williams, John Lowrance, and Ralph Kohler convinced me that Computer Vision was possible; Frank Glazer, Charlie Kohl, and Clif McCormick took me to task for asking how to do Computer Vision; and Daryl Lawton, one day, said, "Look, it's easy, all you have to do is..." and I spent the

next four years doing just that. During that time I relied on the guidance of Debbi Strahman (who started with questions to my answers and ended with answers to my questions) and Joey Griffith (who said, in so many ways, "Well, why don't you just start working on the solution.") and the inspiration of P. Anandan, Randy Ellis, and J. Brian Burns. My thanks to you all.

I will end with those who come first. Many members of my larger family provided support at varying distances. I especially thank my father-in-law for much needed financial support and my mother, because - well, just because. Also, I could not do without my immediate family. My wife, who put up with an ever-increasing absence, not only provided the care and home which every graduate student needs, but was also (variously) copy editor, proofreader, secretary, and psychiatrist. Thank you, Rae Ann! Finally, I offer my greatest thanks to my children, who provided all the reason I needed to continue and finish.

ABSTRACT

Using Object Descriptions in a Schema Network
for Machine Vision
May 1986

Terry Edward Weymouth
B. S., M. S., University of Nebraska
Ph. D., University of Massachusetts
Directed by: Professor Edward Riseman

Computer interpretation of a single static image of a typical natural scene requires the application of a large amount of detailed knowledge. This dissertation explores the information and control structures needed for knowledge-directed interpretation of natural outdoor scenes. A *schema network* represents object descriptions, relations among objects, and control knowledge. Each node of the network, a *schema*, contains both a declarative structure and references to one or more *interpretation strategies*. The declarative portion of the schema describes the composition of an object including the spatial relations of its parts and their possible appearances in an image. The interpretation strategies are object-specific procedures for creating hypotheses of the existence of the object; this procedural representation of control information provides a natural form for expressing the dynamic nature of the image interpretation process.

A *schema instance* is created when a schema is activated either by a top-down request for a goal or by bottom-up detection of key events in the image. Schema instances continually interact with one another, either through a channel set up when a goal is requested or through hypotheses created in a blackboard data structure. Several schema instances can work simultaneously on relatively independent portions of the interpretation, thus exploiting the potential for parallelism. By selectively grouping line and region primitives into descriptions of parts of a scene, the cooperative activities of the schema instances construct the final interpretation network.

The system was tested on six images from four scenes. The parallel execution of the interpretation strategies is simulated and experimental traces are included to illustrate their overlapping activity. The resulting interpretations contain both the association between object structures and image events, as well as three-dimensional descriptions of some of the objects in the scenes.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	v
ABSTRACT	vii
LIST OF FIGURES	xi
LIST OF TABLES	xii

CHAPTER

I. INTRODUCTION	1
Object Representation: A Review of Possible Descriptions	6
Knowledge of Objects in Three-Dimensional Space	11
Knowledge of Images of Objects (Two-Dimensional)	18
Relating Two-Dimensional and Three-Dimensional Knowledge	20
Review of Related Vision Systems	22
Systems Using Three-Dimensional Object Descriptions	23
Systems Using Image-Based and Two-Dimensional Information	28
The VISIONS System	36
Knowledge for Interpretation	39
Discussion of Alternate Control Methods	42
AI and Machine Vision: Schema and Frame	51
Summary	55
II. SYSTEM DESIGN AND ARCHITECTURE	57
An Overview of The Interpretation System	58
Background - The VISIONS System	59
The Images and Segmentation Routines	64
Objects Used in Interpretation	69
Image Interpretation and Expected Interpretation Results	69
Schemas as a Representation of Objects	72
Declarative Representation of an Object	73
An Example Representation: House Roof	74
Three-Dimensional Geometric Structure	78
Image Features	79
Relations	81
Summary	81
Combining Declarative and Procedural Knowledge	82
Activation	84
Schema Instance	88
Communication: Messages and Shared Data	90
Conclusion and Summary	93

III. INTERPRETATION STRATEGIES	95
Basic Interpretation Strategy	97
Exemplar Selection and Feature Matching Extension	103
Grass - Feature-Based Exemplar Selection	104
Grass - Exemplar-Based Hypothesis Extension	113
Sky and Foliage - Additional Exemplar Extension Strategies	116
Extension by Geometric-Guided Construction	120
Shutters - Coalescing Fragments	121
Road - Exemplar and Geometric Constraints	130
Roof (First Strategy) - Exemplar and Geometric Construction	136
Key Feature Matching and Geometric Construction	143
Key Parts and Using the Composition Hierarchy	145
Ground Plane - Collection from STM	146
Outdoor and House Scenes - Basic Part-Whole	149
House and Walls - Key Parts and Using Geometry	150
Context-Initiated Interpretation	157
Using Interpretation Strategies - Summary	161
IV. SCHEMA CONTROLLED INTERPRETATION	163
An Experimental Schema System	166
Considerations in Implementation of Control	166
Our Model of Parallel Computation	170
An Example of a Goal-Directed Interpretation	173
Data-Activated Interpretation	205
Response of Schema Network to Scene Variation	210
Viewpoint Variation	214
Detected Strategy Failure - Roof Interpretation	229
Viewpoint Differentiation and Viewpoint-Specific Strategies	235
Pragmatic Design for Schema Development	243
V. CONCLUSION	251
Review of This Work	251
Future Research	254
Developing Interpretation Strategies	254
Communication Between Schemas	256
Shape	257
Use of Partial Interpretations	258
General Contribution	259
.	
BIBLIOGRAPHY	261

LIST OF FIGURES

1.	Photograph of Outdoor Scene	1
2.	Digitized Outdoor Scene Image	2
3.	Closeup of Roof from Digitization	3
4.	Types of Information Used to Describe an Object	7
5.	Overview of VISIONS System Architecture	59
6.	Multiple Levels of Representation and Processing in VISIONS	61
7.	Representation of Short and Long Term Memory	62
8.	Data Abstraction Processes	66
9.	Object Classes for Interpretation	70
10.	Hand-Generated Example of an Interpretation Network	71
11.	Gable Roof	74
12.	Views of the Roof	76
13.	Network Describing Roof	77
14.	Relation of Schema Instance to a Goal	86
15.	Schema Program and Relation to Schema Instance	89
16.	Histogram-Based Feature Scoring Function	105
17.	Forms of Scoring Functions	107
18.	Feature Histograms with Scoring Functions	110
19.	Combination Function for Grass Exemplar	113
20.	Exemplar Selection and Extension for Grass	114
21.	Sky Interpretation Using Exemplar Selection and Extension	117
22.	Combination Function for Sky	117
23.	Foliage Interpretation Using Exemplar Selection and Extension	118
24.	Combination Function for Foliage	119
25.	Combination Rule for Shutters	122
26.	Interpretation Strategy for Shutters	123
27.	Final Results of Labeling for Shutters	126
28.	Initial Shutter Without Shutter Pair Hypothesis	127
29.	Projection of Shutter Image to House Surface	130
30.	Combination Scores for Road	132
31.	Results for Road Interpretation Strategy	135
32.	Combination Score for Roof	136
33.	Initial Roof Region and Line Data	137
34.	Roof Interpretation Strategy	139
35.	A Camera Model Gives an Approximate Distance	147
36.	Model of House Geometry	151
37.	Relations in Roof Model	152
38.	The House Geometry from the Roof	153

39.	The House-Wall Hypothesis	155
40.	Context-Based Wire Interpretation Strategy	158
41.	Telephone Pole Interpretation Strategy	160
42.	Schema Network	167
43.	Image for Goal-Directed Interpretation Example	174
44.	Image Data for Example Interpretation	175
45.	Time Trace of Schema Activation	179
46.	Relation Between LTM and STM Nodes	181
47.	Results for Grass, Sky, and Foliage Interpretation	183
48.	Road Interpretation Results	187
49.	Roof Interpretation	189
50.	The Relation Between Instances and Schemas	190
51.	House Walls	191
52.	House Geometry	193
53.	Final Interpretation Network	196
54.	Projection of Final Interpretation	198
55.	Projection of House Geometry	200
56.	Projection of Final Interpretation of the House	203
57.	Time Trace of Data-Activated Interpretation	207
58.	Additional Scenes	211
59.	Similar Views of the Same Scene	215
60.	Activity Trace for Similar Images	222
61.	Interpretations of Similar Images	223
62.	Occluded Roof Example	230
63.	Incorrect Interpretation of Roof	234
64.	Interpretation of Occluded Roof Example	236
65.	End-on House Image	240
66.	End-on House Interpretation	244

LIST OF TABLES

1.	Features Used in the Scoring Rules	108
----	--	-----

CHAPTER I

INTRODUCTION

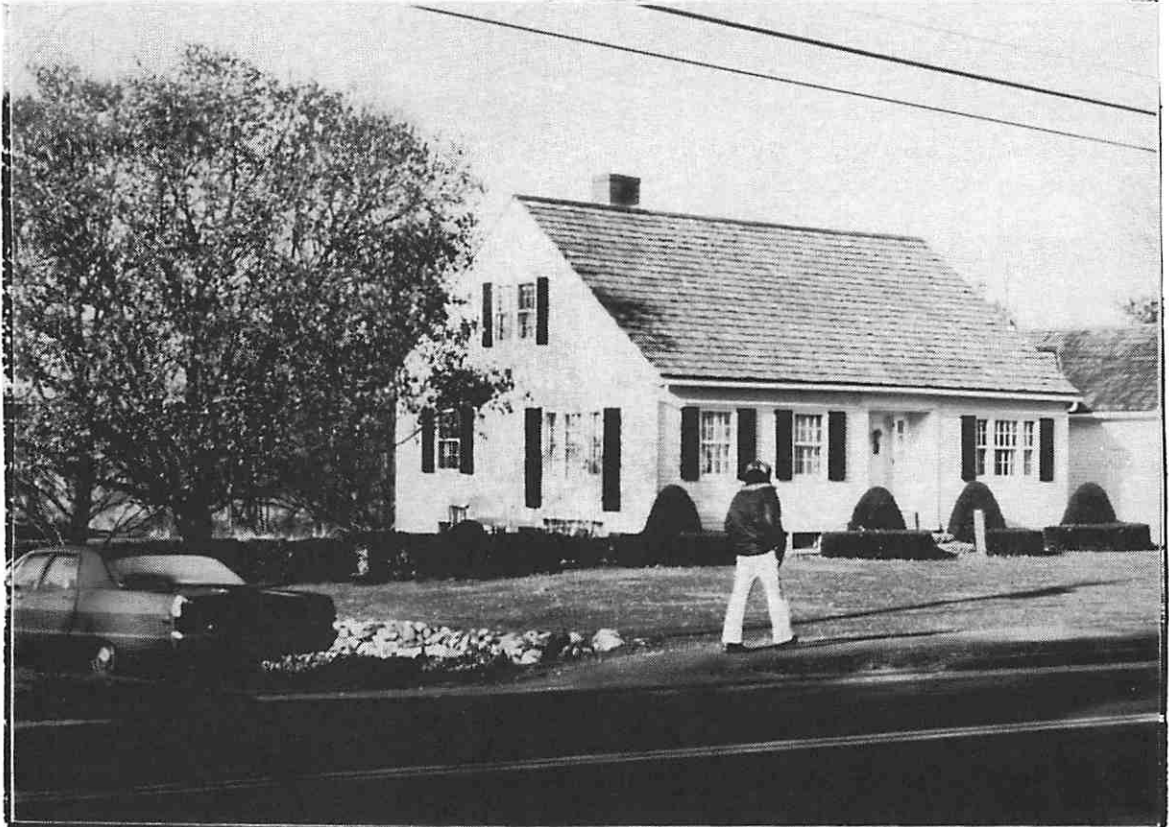


Figure 1. Photograph of Outdoor Scene

This is a typical scene chosen from those used in this dissertation.

In this dissertation we describe a system which interprets photographs of suburban outdoor scenes (such as the one shown in Figure 1). Its development grew out of an investigation into the types of knowledge and control information necessary for effectively interpreting complex images of house scenes. Through the description of this system, we examine two major elements in computer vision: the use of object

models that combine descriptions of geometric structure with procedures for object recognition and the means for coordinating multiple sources of knowledge.



Figure 2. Digitised Outdoor Scene Image

In order to construct an interpretation, a computer vision system must associate image features with objects and scenes. One approach to this problem is to extract from the image as much information about the structure of the scene as possible. This can be done to the extent that such structure is truly independent of subsequent interpretation. For example, consider the roof in Figure 2. It is plausible that the system might be able to recognize its surface as a rectangle before it needed to interpret it as a roof. Further, it might be able to determine the edges of the

rectangle before interpreting that collection of lines as a rectangle. In this approach, a system relies primarily on data from the image and groupings that arise naturally from relations among that data. Such an approach is characterized as *data-driven* or *bottom-up* interpretation.

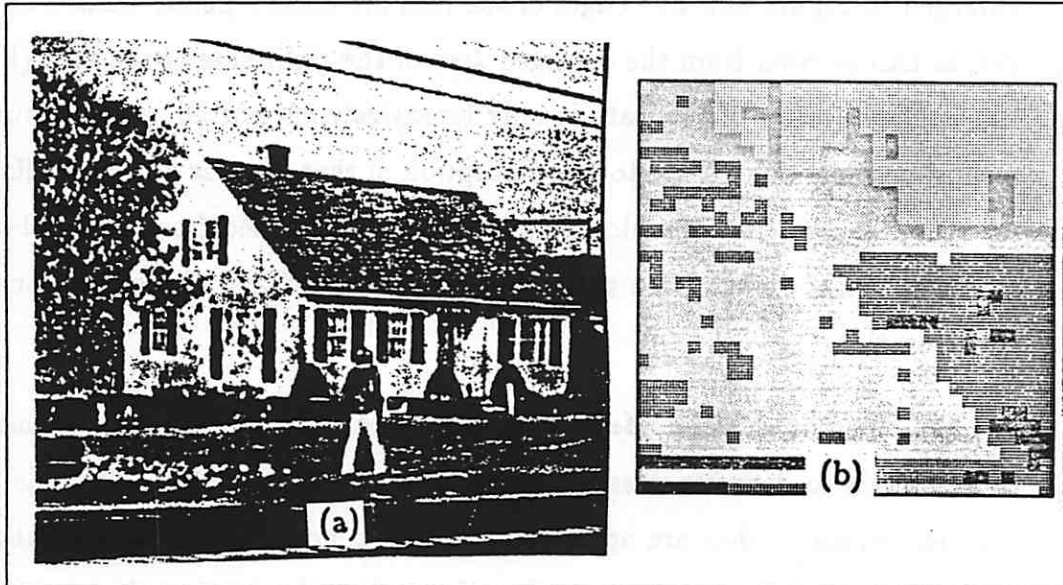


Figure 3. Closeup of Roof from Digitization

Context is used to interpret detail. (a) The boundary on the right side of the roof, though visible in the whole photograph, is not supported by clear evidence from the image; (b) a closeup of the digitized data within the marked square.

The study of data-driven interpretation is advocated by a number of researchers (see [BRA82] for a summary) and has led to an absorbing and intensive investigation of which features can be extracted from the image. For example, several algorithms for the extraction of shape from intensity information, from range data, or from stereo image pairs have been developed. This general approach has encouraged research into models of the imaging process leading to increasingly sophisticated methods of machine vision interpretation.

However, the task of interpreting photographs such as the one digitized in Figure 2 can not rely on data-driven interpretation alone. The primary problem with this approach to image interpretation is that, frequently, the features corresponding to parts of objects (surface edges, in this case) are either not extractable or not even present in the image. Consider the area highlighted in Figure 3a and shown enlarged in Figure 3b. The edges of the roof are clearly perceived as straight lines; yet, as can be seen from the enlarged area of the right side of the roof (Figure 3b), this interpretation of the data is only tentatively supported. Other sources of information must contribute to the perception of that edge as a straight line. We are interested in the case in which those sources are supplied from a model of the roof stored in a knowledge base about the general world and the scene domain under consideration.

Top-down or *model-guided* interpretation relies on object models and propagation of goals to discover missing features. A primary problem with this method is deciding which models are appropriate. We can not afford to test for the presence of every object unless there are only a few objects in the domain, and then only if testing for each of them is not a very expensive operation; neither is true in the case of outdoor scenes.

What is called for is a mixture of data-directed and model-guided interpretation. There are some features that can be more easily extracted from the image by an initial application of data-driven (bottom-up) processing; then knowledge-driven (top-down) processing can be invoked on the basis of this extracted information. The most reliable of these features, those which are strongly supported by the data, can anchor tentative object hypotheses and allow appropriate object models to be selected. Guided by this model, knowledge-driven processing can control the further extraction and grouping of features from the image. Additional features from the data may reinforce the hypothesis; hence, the strengthened hypotheses can

guide the search for even weaker evidence. Thus even initially missed parts can be recognized because of supporting data. Further knowledge-guided processing, including the application of data-directed processes, can be based on previous partial interpretations. Combining model-guided and data-directed processing permits the construction of a description of a scene from the evidence of a single photograph.

The development of the system presented in this dissertation is part of the ongoing research connected with the VISIONS project at the University of Massachusetts [PAR80]. Since its inception this group has emphasized, among other issues, research in the integration of model-guided and data-directed processing [HAN78,HAN83]. Much of the work in this dissertation has been influenced by the conceptual developments of other researchers associated with that project. It is also the case that the programming depends on a software development framework established in the associated laboratory.

This dissertation consists of three major parts. The first, in the remainder of Chapter 1, is a review of related research. The second, in Chapters 2 and 3, describes the details of our system: Chapter 2 concentrates on the system development environment and the role of our system within the VISIONS image understanding framework, and Chapter 3 provides details of the object recognition procedures used. The final part, in Chapters 4 and 5, describes the results of running the interpretation system on a set of suburban outdoor scene images. Chapter 4 exhibits specific results, while Chapter 5 presents a more general analysis.

To begin this study, Chapter 1 concentrates on related research. Section 1 reviews methods for representing information about objects; Section 2 reviews computer vision systems that use object models for interpretation; and Section 3 contrasts several methods of combining information to be used for control. Finally,

Section 4 describes the background of the particular method that we chose for representing information and control, which is the *schema*, a frame-like structure with attached processes.

1. Object Representation: A Review of Possible Descriptions

There are many types of information that can be used for object recognition. These include the physical attributes of the object, such as shape, size, color, and the relative orientation and placement of the parts of an object. Other useful types of information include specialized knowledge about how the projection of an object will appear in a digitized image, and how objects are related to other objects within a setting or scene. In this section, as we review related research, we will discuss what these various types of information are and how they are represented in object descriptions.

We begin our discussion by presenting a categorization of the types of knowledge used to describe an object. The framework, presented in Figure 4, divides the information into five related networks: an "is-a" hierarchy, a "part-of" hierarchy, a network of three-dimensional geometry, a resolution hierarchy, and (for each class of viewpoints) a network of image-based, two-dimensional spatial relations. Each network is simply a data structure in which entities are represented as labels or nodes and relations are represented as links between those nodes. Embedded in these structures are these additional relations: constraints on color and texture, indications of surface markings, constraints on relative position, and symbolic spatial relations.

The first hierarchy is a natural consequence of object classification. Every object belongs to several classes of objects, which can usually be arranged in a hierarchy of inclusion (called a subclass, "is-a," or specialization hierarchy). For example, a Volkswagen is a type of car, which is a type of vehicle, which is a type of mechanical

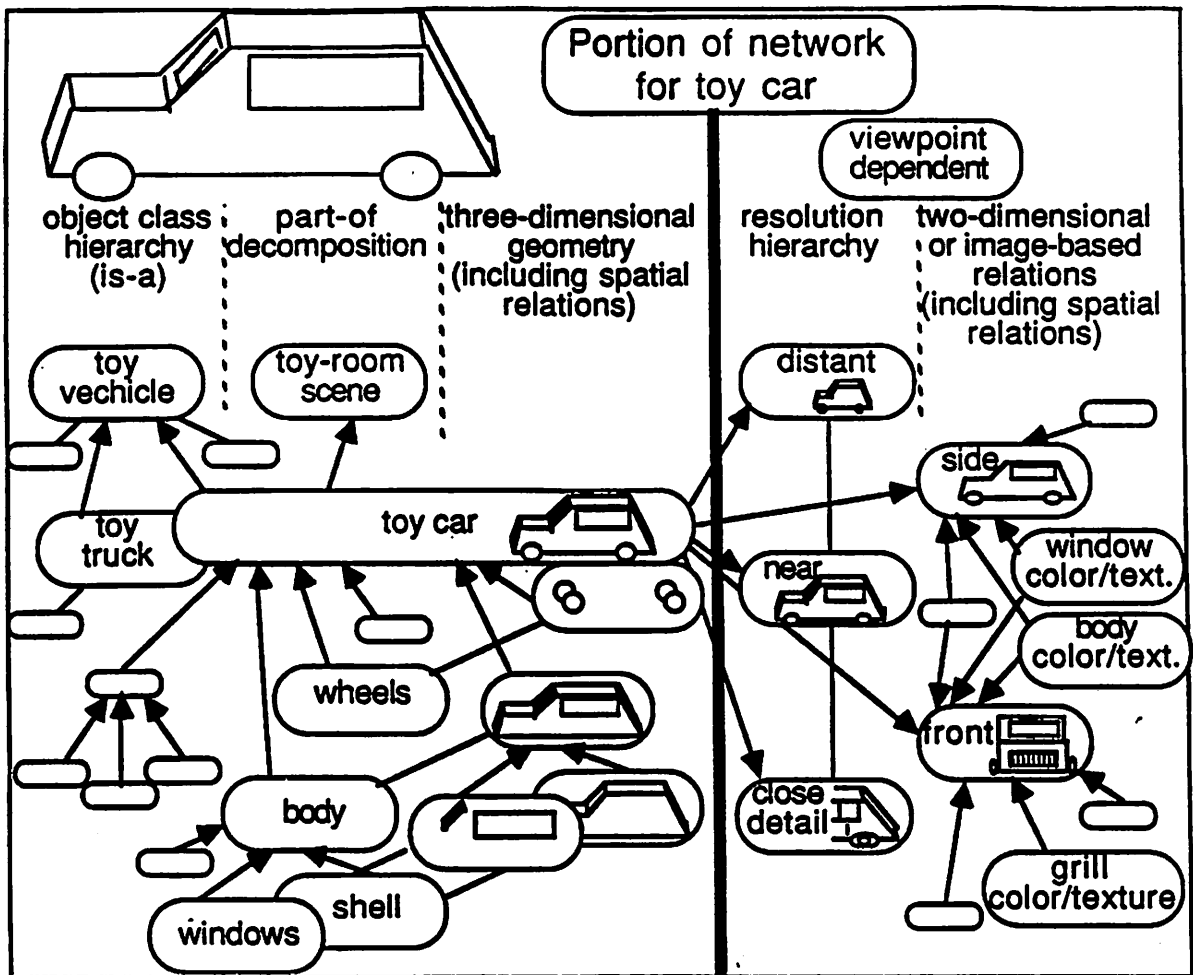


Figure 4. Types of Information Used to Describe an Object

A network of objects, object features and image features is used to represent information for image interpretation. Conceptually, the network can be divided into five overlapping sub-networks: the relational hierarchies (is-a and part-of), the network of three-dimensional spatial relations, and geometric information represent view independent information; the viewpoint dependent knowledge is represented in a resolution hierarchy and a network of sets of two-dimensional (image-based) relations. In the depiction of the geometric information (2D and 3D), the small drawings stand for a sub-network containing shape descriptions and relative positional information; geometric information can also include precise sub-part placement and indication of common sub-parts. Additional relations and descriptions include constraints on surface color and texture.

object. By building a network of these relations, we form a specialization hierarchy. The specialization hierarchy is useful in object recognition when it can be augmented with those features that distinguish one object class from another. Thus, in addition to representing the object class as a label, the node in a network is often augmented with a set of parameters and values which describe the object. Such augmented nodes are called *frames* or *schemas*.

A schema (or frame) is a data structure for the network node consisting of an object label plus a set of name-value pairs that describes the object. The name-value pairs are called slots. The names are the labels of attributes of objects in general (or at least of a general class of objects) and the values are the expected or default values associated with that particular attribute of the particular object class. For example, a car has four wheels, a bicycle has two, and they are both vehicles. In a specialization hierarchy vehicles would be represented by a frame with two slots (among others): one for number-of-wheels, and one for pointers in the specialization hierarchy. For an undifferentiated vehicle the number-of-wheels slot would have an undefined value; but that slot would also occur in the bicycle and car frames and the values would be set appropriately. In addition to static values, frame slots are often filled with parameterized references to functions. In such a case, the frame is said to have *procedural attachment*. Frames (schemas) can be used in all of the networks described.

The second type of relation is the "part-of" relation, also referred to as the decomposition or composition relation. These relations also form a hierarchy. For example, car is composed of a body, shell, and wheels; the shell consists of windshield, doors, fenders, hood, etc. Recognition of any part indicates the likely presence of the whole, interpretation of the whole implies searching for the parts, and the relations among the parts can be used to confirm the recognition and constrain the search.

It is not enough to see the parts of an object. Recognition consists of discovering image features in their correct spatial relations; thus, describing object geometry is also useful. Objects occupy volume and they present visible surfaces. They articulate in specific ways. Their surfaces curve and fold to present edges and vertices. These geometric features and their spatial relations can also be used to describe an object. For some types of objects there is a direct correlation between the geometric features of the object and the features that would be expected in an image of that object.

If an object is being viewed from a particular point of view, there will be characteristics of its image that are typical of that view and others like it. Parts of the object may not be distinctly visible because they are too small or occluded. When a vision system is likely to encounter an object in one of a few known positions, then the process for recognizing that object can take advantage of this knowledge and use viewpoint-specific image features. Thus, it has a computational advantage over processes that must derive the expected image features from the general three-dimensional description of the object in space. When viewed from a known range of viewpoints, the object description can include the image characteristics as features in a projection, with relations among them forming a network. Since this network describes the object from a particular class of viewpoints, it separates the relations into subnetworks, each of which we will call a *view*.

Each view represents a class of viewpoints having the distinction that a small change in position causes no significant change in the perceived relations. For example, if you know that you are looking at the side of a toy car and are a moderate distance from it, you will expect to see in the image the wheels as distinct circular or approximately circular objects and the car body in clear profile. Changing your vantage point slightly does not affect what is visible or the perceived shapes to any great extent. However, a large change in viewpoint will cause a significant change

in perceived shape. Thus, when you shift to look at the front of the car the features of expected image size, shape, and distinctness of surface markings clearly vary. On the other hand, the features of color and size relative to nearby objects, for example, remain unchanged and can be represented by relations not included in the grouping for a particular view.

The previous discussion suggests the utility of two networks: a resolution hierarchy and a network of image-based, two-dimensional, viewpoint-specific relations. The resolution hierarchy encompasses those relations and descriptions that are affected by the distance from the viewer to the object. When an object is to be viewed from widely differing distances, the information appropriate to the approximate viewing distance can be more readily accessed by separating descriptions of overall shape from descriptions of detail and relating them in a resolution hierarchy. From a sufficiently great distance the parts of any object will appear indistinct, perhaps having one or two clear features. As the viewing distance to the object is reduced, the details of the major parts of the object become clear (the wheels and windows on the car, for example). Finally, at a close viewing distance very fine detail is visible (door handle, chrome trim, etc.).

Each of the networks described so far represents some aspect of the object in space. Even the resolution hierarchy, which relates spatial detail to visibility, can be based on a three-dimensional description of the object and its parts. In contrast to dealing with the three-dimensional form of an object, it is also useful for a vision system to have available a two-dimensional image-based model of the object. Because it is frequently possible to describe objects solely in terms of their images, interpretation processes can be developed while avoiding the complex issues which arise when attempting to use general descriptions of three-dimensional shape.

The network of image-based features and relations in each *view* facilitates interpretation based on image information. When a class of possible viewpoints for an object can be determined, then the image features can be used to recognize the object. Observe that for some image features (e.g., color and texture) the range of possible viewpoints may be very large. However, even when dealing with geometric information, viewpoint-specific information can be helpful. For example, see the discussion of the house strategy in Chapter 3 and the last experiment described in Chapter 4.

Since all of the networks described above share nodes and relations, they are really aspects of a larger network. The various hierarchies and other networks intertwine to form a description of a set of objects. Thus, for example, a recognition process that starts using viewpoint-specific information can eventually use the three-dimensional geometric description or the compositional hierarchy to further the interpretation. The relations among aspects of an object are as important as the relations among objects.

As a foundation for the review of related research, we make a distinction between information in an object representation that pertains to the object as it appears in the physical world, and information that pertains to features of two-dimensional projections or images of the object. In what follows, we will first discuss the types of three-dimensional representations that are commonly used; then we will review types of two-dimensional, image-based representations; and finally, we describe the relations between these two classes of information and how those relations can be exploited for image understanding.

1.1 Knowledge of Objects in Three-Dimensional Space

Objects are three-dimensional. Thus, describing the three-dimensional structure of an object is one of the more obvious ways of representing it. Ballard and

Brown [BAL82, Chapter 9] summarize methods for representing the three-dimensional geometry of objects. One way to represent an object is to describe the volume or the "space" occupied by the object. Usually, this is done with a set of primitive solids and groupings on those primitives. The simplest type of volumetric object description is based on a single primitive form. Cubes or other simple volumes are combined so as to describe the space filled by the object. For example, Badler [BAD79] used spheres as a space-filling primitive, with an object being described as a set of spheres (possibly of different sizes). This representation has the advantage of being very simple. For a sphere of arbitrary size and position only four numbers are needed: one for the radius and three for the position of the center in space. Additional numbers may be needed if ranges of values are desired; to represent volumes at higher resolution, smaller spheres are used. An economy in the number of primitives required can be achieved through mixing spheres of different sizes. However, a disadvantage of this type of representation is that for objects of complex shape a large number of primitives may be needed. For example, describing a telephone with a reasonable amount of detail would require a large number of small spheres. To get around this problem, we can either increase the number of types of primitives or allow primitives of greater complexity.

One method of representing volume which uses a greater number of primitives is constructive solid geometry (for example, see [VOE78] on the PADL system). Primitives in this type of representation are simple convex solids, which are combined by set operations such as "join", "intersect", and "subtract" (for union, intersection, and difference). This style of representation has been used in design and display and has also been used in image understanding for recognition of simple combinations of primitive solid shapes ([ROB64]).

A commonly accepted primitive of greater complexity is the generalized cylinder. It consists of a curve in space called an axis and a function that describes the

cross section of the volume as it varies with position along the axis, in the plane perpendicular to the axis. In the next section, where we discuss machine vision systems, we will examine the work of Brooks [BRO81], in which generalized cylinders describe objects for image interpretation.

While volumetric representations such as generalized cylinders and constructive solid geometry describe the volume of the object, the generality they afford is not always necessary. A problem common to these representations is that they do not correspond to image features in a way that leads easily to matching. It is necessary to derive "features" of the representation that can then be matched with the features in the image. For example, characteristics of the object surface are not directly represented and must be derived when needed. As an alternative, the geometry of an object can be represented as a composition of joined surfaces, called patches. Such patches can be complex, like the parametric patches, or simple, like the polygon patches. Patch representations, though perhaps not appropriate when a description of volume is necessary, suffice when only a description of the surface of the object is needed.

The simplest type of patch is a polygon in an oriented plane, represented as lists of points in space. This type of representation is frequently displayed as the lines between the points, called a "wire frame" display. A more general patch representation uses a parametric function of the surface. The Bezier and B-spline surfaces are two such parametric patch representations. See [FAU79] or [ROG76] for a discussion of parametric patches and [YOR80] and [YOR81] for a discussion of how surface patch representations might be used in computer vision.

For some problems in machine vision, a representation need not always describe the details of the three-dimensional geometry of the object. Ikeuchi [IKE80] suggested using a Gaussian sphere, in which the object is described by the distribution

of surface orientations per unit surface area (a histogram of the surface normal on the unit sphere), without regard to surface position. This forms a "signature" for the object which can be related to extracted surface orientation (see, for example, Horn [HOR77]).

Some systems use an explicit representation of spatial relations to augment geometric descriptions. These relations can be expressed as precise spatial relations (e.g., the angle between two surfaces) in the form of constraints on spatial variables (e.g., bounds on the range that a particular value may take), or symbolic relations (such as "next to" or "above"). Symbolic spatial relations have their own problems, such as the difficulty of constructing reasonable computational models for relations such as "above;" however, they can be useful.

Other characteristics of the object can be used in the case where a complete geometric description is not always feasible. To illustrate this, consider a description of a tree. For some purposes, a description of approximate geometry would be sufficient; that is, a tree might be modeled as a cylinder (the trunk) joined to a sphere (the crown). If very rough approximations of shape and location were all that were needed, this style of description would suffice. However, there is no easy way to model the geometry of a given tree with enough detail so that the texture and color of the precise patterns of leaves could be derived from the representation.

It is clear that the parts of a tree have a characteristic color and texture, but how are these to be represented? One approach is to represent the color and texture as characteristics of a surface. This is a standard computer graphics approach [FOL82]. For example, detail can be represented by describing the surface color or texture as a repeated pattern using a function or a lookup table with interpolation [BLI77], an approach which is sufficient for many objects. As an illustration that is easily visualized, an orange could be represented as a sphere (three-dimensional

geometric shape) with the surface color variation and surface texture represented as values in a lookup table indexed by the spherical coordinates of the surface.

A more intriguing approach to the problem posed by the tree example is to adapt methods of representation being developed for computer graphics. In their attempts to generate increasingly more realistic images, researchers like Smith [SMI84], Kawaguchi [KAW82], and Reeves [REV83] are developing representations of objects capable of generating images that are fractal in nature or have other random but constrained characteristics. Whether such representations would be useful in recognizing objects remains an open question. The success that these researchers have had in succinctly expressing the characteristics of very complex object forms (such as trees) leads to speculation that the description of natural objects, for the task of recognition — if there is ever to be a general description of such objects — will have information of a similar nature. (See [PEN83] for an example of using fractal information in the analysis of restricted types of images.)

In general, the problems of representing objects that do not have a clearly defined shape (clouds) or those that have distinct shape requiring far too much detail to model completely (trees) have resisted any elegant solution. The most common approach, in both computer graphics and computer vision, has been either to model these objects in two dimensions (see second part of this section) or to provide hooks in the general three-dimensional representation for routines to generate their images (in computer graphics) or for specialized programs to recognize the objects (in computer vision).

In addition to precise models of the three-dimensional shape of an object, symbolic relations are used to describe characteristics of objects. The symbolic representation of spatial relations was mentioned above in the discussion of representations

of geometric information; to these we add three types of inter-object relations: composition, specialization, and resolution. Each of these relations forms a hierarchy of objects and object parts.

Note that the distinction between objects and object parts is somewhat arbitrary and determined by context. In speaking of a box, the lid is part of the object; but the knob on the lid is part of an object, the lid of the box. Thus, when we speak of "relations between objects" we mean to include those between the parts of an object, between an object and its parts, and between objects, unless otherwise indicated.

Hierarchical relations provide an interpretation system with paths of logical inference to follow when interpreting a scene or collection of objects. As mentioned before, three types of hierarchies are considered: the compositional ("part-of") hierarchy, the specialization ("is-a" or subclass) hierarchy, and the resolution hierarchy.

A compositional hierarchy is built on the relations of parts to the whole. If, for example, a system needed to know that a child's block has six faces arranged in three mutually orthogonal parallel pairs, a compositional hierarchy could represent this directly. Of course, this information could be derived from a general geometric description, but it might be more efficient to have it represented explicitly. Thus, we can represent the pair of faces as a part of the block and each face of the pair as a part of the pair.

A specialization hierarchy is used when describing one object as a type of another. For example, in a specialization hierarchy for a toy block we might have a general description of the block as a cube, a more specific description of the block as one that was small and with letters and numbers on the faces, and finally a description of a particular type of block with the letter A on one face. Brachman [BRA76] points out that there are really two types of specialization. He calls the cases that we have described "para-individuals," from parameterized individualization; they

are different from the case of a particular block, e.g., Tommy Allen's block that he bit on October 8, with an A on the face.

This distinction between specialization and individual instances can be used to an advantage by vision systems. In many applications, images of identical objects must be distinguished from multiple images of the same object. The first is a para-individual of the object type; the second is a particular object. It is reasonable to expect that a system would have para-individual descriptions in its static (or long term) memory and descriptions of particular objects in its dynamic (or short term) memory in the description of a given scene. However, Brachman makes the point that we may also have a need for storing the description of the particular object in the static memory.

The third type of hierarchy, based on resolution, is useful for relating representations of differing detail. For example, we might represent a house at three levels of resolution. From a distance, it would appear as a roughly boxlike structure with a clearly distinguishable roof and corners, but without detailed structure. At a medium resolution, the roof and walls would be represented with their major landmarks, structures such as the window frames and doors; and, at fine resolution, details would be added: the representation would include doorknobs, eaves, gutters, and television antennas. (See [CRO82] for an example of a resolution hierarchy used in computer graphics.)

The three types of relational hierarchies and the spatial relations between objects are part of the information about objects as they occur in space and can be used to guide interpretation. We have looked at several types of three-dimensional knowledge that we might find in an object representation: descriptions of object geometry, descriptions of object surfaces, and descriptions of the relations between objects expressed through hierarchies.

1.2 Knowledge of the Images of Objects (Two-Dimensional)

We now turn, in our discussion, to the representation of knowledge that relates to views of the object. The photographic process is a projection of the visible surfaces of objects in a physical setting (possibly having complex lighting properties) to a two-dimensional image. The image interpretation problem involves inverting that projection. One approach is to try to determine the appropriate three-dimensional structure from the image. This is a complex and ill-defined task with no unique solution. Rather than matching an image to a three-dimensional model, it is often feasible to store information about the possible appearances of the object. Even information about one particular, but prototypical, projection of the object may be helpful. With this type of information, the image of the object can be matched directly to the description.

Representations of two-dimensional knowledge can extend an object model by describing features of the images of the object. This use of image features relies on the assumption that the image of the object, and measurements made from it, are a reasonable approximation of an ideal projection of a (possibly unknown) full three-dimensional representation. An example occurs in the analysis of aerial photographs, where a building might be represented as a "rectangle with a shadow" (see [NAG80]). Such characterizations of object images have also been shown to be useful in areas such as x-ray analysis and other medical applications. The two-dimensional nature of information about expected image data and relations makes the task of matching object description to image easier. Image-based descriptions are most frequently used when a general description of an object is not available or when, even if such a general description is available, matching the three-dimensional description to the image presents problems.

One method for representing two-dimensional, image-based, geometric information consists of a description of the boundary: typical examples ([BAL82]) are chain

coding, polygons, conic-sections, and B-splines. Another is a two-dimensional correlate to generalized cylinders, which is used to describe the two-dimensional area covered by an object; the image area is described by a curve along its axis and a width-function that sweeps out the area as an integral of the perpendicular distance from the axis. The axis can also be used alone as an abstraction of the area shape. Other characterizations of shape are possible: for example, the number of corners or number of sides, the fractal dimension of the boundary, or such summary measurements as compactness. (See [BAL82] for a more detailed discussion of two-dimensional shape descriptions.)

The color and texture of the object's surface are important characteristics. For a particular view, they can be represented by the color and texture of the corresponding area in the image. Color and texture can be measured as statistics of image regions. In this case, the object description is an empirically derived description of a prototypical region for the object and this prototype can be matched to regions in the image. The analysis of aerial photographs often uses object models of this nature. Thus, the representation of a river might translate roughly as an area that is a "dark winding strip."

In addition to describing the possible images of the object, it is also useful to have information about the relations among or between objects in the image. As with the three-dimensional knowledge, both metric and symbolic relations are possible. Spatial relations include numeric relations such as distance-between-objects, angle-between-lines and relative-size, and symbolic relations such as attached-to, to-the-left-of and next-to. Another class of symbolic relations is that which relates color and texture of regions, such as darker-than and same-color-as. For example, without knowing the color of a house window shutter it would be helpful to know that each image area designated as a shutter would probably have similar color.

1.3 Relating Two-Dimensional and Three-Dimensional Knowledge

Having a complete three-dimensional description of an object does not solve the problem of image interpretation. Since the image of an object is a projection of that object to a plane, there remains the problem of figuring out where the object is in the image, what position it is in, and how it was projected to the image. There are, in general, three ways to match object descriptions to image data:

- match three-dimensional object models to three-dimensional information derived from the image
- match two-dimensional (partial) object models derived from three-dimensional models to the two-dimensional image data
- match two-dimensional object models to two-dimensional image data.

Extracting three-dimensional information from the image to match with three-dimensional object models is proposed in [HOR77]. It is also the method used by image understanding systems that have detailed depth data from stereo or directly from range measurements. However, it is sometimes difficult (if not impossible) to derive the three-dimensional shape of the objects directly from the two-dimensional image data.

If we use a procedure that produces, from the three-dimensional model, a corresponding two-dimensional model, we can then match that two-dimensional description to two-dimensional abstractions of image events. For example, Brooks [BRO81] represents objects as generalized cylinders. A projection of this object description, the contour of the occluding edges called a "ribbon," is used to predict the appearance of the objects in the image. These are then matched to corresponding groupings of lines that have been extracted from the image.

We can describe the object in terms of its possible two-dimensional views. Some qualities of the object image are invariant in a large number of possible images. If we can characterize a set of viewpoints by an invariant feature that we can measure in the image, and do the same thing for any other set of viewpoints that have such invariants, then the matching process can be simplified. When any one of the features is detected in the image, it indicates which set of viewpoints to use. Also, the detection of view-invariant features localizes the position of the object. Minsky [MIN75] suggested this general idea as a means of representing spatial knowledge; further, he suggested that intermediate views might be obtained by using transformations between the two-dimensional views. Using descriptions of views to represent the possible projections of the object has the advantage that correspondences between the three-dimensional description and the two-dimensional projection are explicitly represented, thus bypassing the need to compute the projection from a three-dimensional model. This is essential when such computations are either too costly or impossible.

In summary, we see that there are many options for describing objects for object recognition. While representing three-dimensional geometry allows a general description of objects, representing the typical two-dimensional projections of the objects and matching these to image-based features bypasses many of the problems of three-dimensional matching. Further, by characterizing how the objects appear in images we can describe objects that are too complex to describe in terms of three-dimensional geometry. It would appear that the best approach is to enable the system to use both three-dimensional object models (when possible) combined with two-dimensional object descriptions that include the use of image-based information.

2. Review of Related Vision Systems

We now turn to a discussion of computer vision systems that use descriptions of objects, or object models, in image interpretation. We are interested in examining how the types of object-related information described in the previous section are applied in computer vision systems. This is not an exhaustive review; it is a summary of some working systems which make effective use of information about objects. (Many of the systems discussed here are described in Binford's "Survey of Model Based Image Analysis Systems" [BIN82].)

At issue are the relations between representation and control. Control is the process of choosing what actions the system will take. In examining the relationship between representation and control in the systems reviewed, we attempt to answer three questions:

- What types of knowledge are used by the system? Are there other types of knowledge, aside from the knowledge related to objects, that researchers have found important?
- How does each system represent knowledge? What explicit choices for representation were made by the researcher? What effect do the choices for representation have on system design?
- What method of control is used? What heuristics have been added to limit search and curtail processing?

Some systems are based on representation of the three-dimensional geometry of objects, while others rely more heavily on image-based knowledge. We shall use this distinction to divide the systems we review, beginning with those systems that use an explicit representation of the general three-dimensional shape of the object.

2.1 Systems Using Three-Dimensional Object Descriptions

All of the systems discussed in this section recognize objects by matching lines in the image to the edges of objects derived from a general representation of three-dimensional object geometry. For such systems, the shape and structure of each object is described in an object model using a general three-dimensional method of description. Each system has an associated method for matching the object model to corresponding image features. However, these systems vary widely in the results they achieve and in the generality of their representations and methods of matching. Not surprisingly, the systems developed more recently are more general.

The system by Roberts [ROB64] demonstrates that image interpretation can be guided using knowledge of object geometry. His system performed object recognition on a scene composed of a single object, having a simple geometry, viewed under tightly controlled lighting conditions (a type of recognition problem referred to as a "blocks world" problem). The system uses a polygon facet representation of objects augmented with a graph expressing the topological relations of edges and vertices.

The topological graph and the polygon based description are directly related to the stages of the matching procedure. Matching topological relations first, the system determines a correspondence between model points and image points for the subsequent computation of a fit of the model to the image. The image points and the corresponding model points form a set of simultaneous equations for the unknown translation, rotation, and scaling parameters, which (generally) is overconstrained. The system solves these equations for the unknowns using a least-squares approximation, selecting the object with the lowest error in the least-squares fit between data and model points as the interpretation of the image.

In addition to knowledge about object geometry, Roberts's system uses knowledge about the interpretation process. Specifically, the ordering of steps in his

algorithm implies knowledge about the processes for extracting features, matching, and error recovery. For example, a bad image vertex value is removed from the matching computation when the error contribution from the vertex is exceptionally high. There are two additional examples. In the first, the procedure that extracts the lines from the image corresponding to object edges involves several thresholds; thus, it represents knowledge about how image intensity discontinuities correspond to object edges (e.g., expected contrast and continuity). In the second, the procedure that determines the overall scaling of the object is based on distance from the camera as estimated by the distance from the bottom of the image of the polyhedron to the bottom of the image frame. Thus, it represents knowledge about perspective. In these examples knowledge is represented directly in procedures.

The system developed by Roberts relies primarily on knowledge of object geometry. Falk [FAL72], Turner [TUR74], and Shirai [SHI78] developed and extended the idea of using a set of geometric primitives, and the constraints implied by them, to match objects in a scene. Although this is a common approach, all these studies confine their experiments to worlds consisting of polyhedral objects.

The system designed by Paul [PAU77] interprets images of an articulated wooden marionette in various positions. Again, as with the work of Roberts, the lighting conditions are tightly controlled and the image is of only one object. However, as this is a more complex object, Paul's focus shifts to the problem of determining the relative position of the torso and limbs of the marionette. Thus, his system relies on knowledge about the possible relations among the parts of the object; for example, the model includes knowledge about the possible angles between the limbs and the torso. Paul's system represents the parts of the marionette as a set of joined solids with restricted attachment properties. Specifically, the knowledge about the marionette body parts and their relations is represented as a hierarchical graph of part-whole relations, with each body part (each limb, for example) represented by

a wire-frame model. Thus, knowledge of the possible positions of the marionette is derived from a single declarative description.

In addition to the model of the object, Paul's system relies on procedurally encoded knowledge about how the parts might appear in the image. Naturally, there is a direct relation between the types of features the system can extract from the image and the description of the image features in the object model. Because the system relies on image data routines which are better at finding the straight lines corresponding to edges that are not near any vertex or body joint, grouping algorithms look for evidence of related groups of straight lines. Also, occluding contours are especially useful, both because they are frequently detected (due to a smooth background), and because they contain strong clues about the positions of limbs. Knowledge about which features to match and the order in which to best perform the matches is encoded in programs - a procedural representation as distinct from the declarative representation of the marionette.

Additional heuristic control knowledge (such as which image features to match first and with which parts of the model) is also represented as part of a procedure in a searching program. Like Roberts's system, Paul's system determines a correspondence as the first step of interpretation. To do this, the system limits the search for possible groupings of straight lines and the ways in which they can be matched to the model by considering information about the attachment of parts of the puppet. Constraints on the placement of parts, from the information in the three-dimensional description of the puppet, imply two-dimensional adjacency relations in groups of straight lines in the image. These limiting, two-dimensional relations, in turn, constrain the labeling of the groups of lines. Paul's system builds a description of the image by heuristically guiding the process of matching of an object model to the image features.

Later developments have been directed towards building increasingly general representations of the object geometry for image interpretation. The system developed by Brooks [BRO81] relies on knowledge about geometry of the object and about inter-object relations. Each object is modeled as sets of generalized cylinders. Geometric relations between object parts, such as the possible angles between a wing and the body of an airplane, are represented as parameterized relations. Objects can also be associated by part-whole relations (composition) and class-subclass relations (specialization). These two relations form hierarchies, which are augmented by an attachment relation. Each relation can be constrained by having algebraic inequalities on parameters of the object description or the relation. For example, an aircraft would be represented as a body composed of parts including two wings with constraints placed on the angles between each wing and the body. All constraint-expressions having the same parameter variables participate in an implied relation among them, in that they reference the same global variable name. This network of constrained object descriptions constitutes the description of the objects.

The programs for matching data to object representations contain additional knowledge. For example, in building the description of an image the system proceeds from specific to general and from data to model when initially matching objects to data. However, when the system attempts to merge two partial interpretations (by merging their interpretation graphs), it uses the constraint relations to make choices. Further, the type of image feature matched depends directly on what two-dimensional attributes can be derived from the object model. Additional heuristic knowledge about the types of image features determines how matching should take place. The image feature abstraction process produces *ribbons*, defined as the possible projections of a particular type of generalized cylinder. The interpretation process matches the object representation to the image by searching

among groupings of ribbons to determine a correspondence of generalized cylinders to ribbons resulting in a consistent set of constraints.

In the experimental part of his work, Brooks applied this method to a set of aerial photographs of an airport and was able to recognize the position and orientation of several airplanes on the ground. However, some airplanes were missed because the system was unable to generate ribbons corresponding to key parts. Binford, in his review of Brooks's work [BIN82], argues that the interpretation results involving missed objects followed from the lack of good feature extraction processes. Despite this, Brooks's system did do a creditable job with the ribbons that were found.

The three systems developed by Roberts, Paul, and Brooks illustrate the advance of models based on object geometry. They also demonstrate some of the limits of such research. Roberts's system profits by the choices of the domain for which it was developed, by the simplicity of its object models, and by the specificity of its matching techniques. Paul's system also takes advantage of a judicious choice of lighting and the approximation of the object as polygonal surfaces. Although the object in Paul's system is more complex than those that Roberts used, they both succeeded at solving recognition problems for what are very limited domains. In contrast, the system developed by Brooks interprets complex images using a far more general model and much more general matching procedures. While in principle Brooks's method of constraint propagation could be used in recognizing various photographs of a three-dimensional scene, two observations are relevant. First, with respect to the interpretation of object images, his proposal has not been fully tested. Generally, aerial photographs have a two-dimensional quality that would not fully exercise the abilities of a system to interpret three-dimensional objects. Viewing an object from above and at a distance permits the use of simplifying assumptions

about the method of projection, and about the presence of object contours. Second, it seems that information of three-dimensional shape and size, by itself, is not sufficient in many domains. As a simple example, one wonders whether Brooks's system might have identified more airplanes if it had used knowledge about the possible contrast and relative intensity between the body of an airplane and the runway. Given the current technology, developers of general vision systems have much to gain from integrating knowledge from more than one source.

2.2 Systems Using Image-Based and Two-Dimensional Information

In addition to three-dimensional geometry, objects have surface characteristics such as color and texture which produce corresponding image characteristics. In this section we review those systems that concentrate on the use of image-features related to surface color and texture, and two-dimensional spatial relations. Understandably, the systems that use knowledge about how objects might appear in the image, from a *specific* viewpoint, are far more common than those that attempt to use general image-based or two-dimensional descriptions of objects for a general viewpoint or which attempt to determine the viewpoint from the image. However, even when interpreting a known view of a known scene, the task of object recognition presents problems.

Barrow and Tenenbaum, in [BAR76], describe a system that uses image-based information for interpretation. In their system, knowledge about objects is represented as a set of constraints on relations between values measured in the regions of an image. Object labels are associated with regions from image-based measurements of color and texture. Consistent labelings are determined from the constraints given by spatial relations and relations comparing region color or texture. Thus, for example, a bright area above a dark textured area, in an outdoor scene, might be sky above foliage. A relaxation labeling algorithm finds the most consistent labeling of the image regions by propagating the effects of the set of constraints. When

unambiguous labelings can not be found, a heuristic search selects, from among the most likely sets of labelings, the one that is most consistent.

Some knowledge is represented procedurally. For example, the procedure which labels regions as homogeneous is based on a measure of the variation in intensity within the regions. It contains knowledge about the imaging process (as, for example, the expected level of noise). Such procedures represent knowledge about expected image features and, by selection of labels, exert some control over the interpretation process. Further control comes from the application of relaxation labeling. A similar approach is used by Bajcsy and Joshi [BAJ78] and by Sloan [SLO77].

Tenenbaum and Barrow, in [TEN76], examine the combining of a description of the three-dimensional geometry with information about the constraints on image-based feature values. They show that a three-dimensional model of a motor casing can be used to guide interpretation to a picture of the casing; information about the intensity relations among regions of the segmentation helps localize the image of the motor within the picture. The three-dimensional model is then used to identify visible bounding lines. This is a specific example which illustrates the usefulness of combining image-based information with three-dimensional information.

Other researchers (e.g., Barrow and Popplestone [BAR71], Ballard, Brown, and Feldman [BAL78], and Levine [LEV81]) use image-based descriptions to interpret photographs. These descriptions include such information as two-dimensional geometry, two-dimensional spatial relations, constraints on values of region color and texture, and similarity relations.

The general problem with using image-based measurements is to capture those that are relatively invariant. Using features of the images of objects and of models of their two-dimensional projections reduces the complexity of matching the object

descriptions to the image. Unfortunately, this ease of recognition comes at the cost of a loss of generality. It is comparatively easy to build a general system to interpret only one photograph; the danger is, however, that such a system relies on image-specific knowledge in such a way that other photographs of the same scene would not be as easily interpreted.

The system developed by Ohta [OHT80] for understanding images of buildings in outdoor settings also makes use of image-based information. In addition to knowledge about the expected color and texture of regions, his system uses knowledge of two-dimensional spatial relations. For example, the system looks for images of windows with size and position relations in the image that suggest a common vanishing point. Ohta's system also has a part-whole (compositional) hierarchy that determines which objects should be interpreted first. His interpretation process proceeds from major object to detail and is controlled by relations expressed in a production system architecture. These relations are expressed as symbolic rules, of the nature of "when I see a large blue region, hypothesize that it is the sky;" that is, they are pattern-action pairs. The assemblage of these pattern-action pairs constitutes the knowledge base for the interpretation process.

In Ohta's system, some of the information pertaining to particular objects is embedded in procedures. These procedures can be activated either during pattern matching, or as part of the "action" of a pattern-action pair. The information in the patterns is contained in predicate-like procedures that make comparisons or test values against thresholds. In the actions there are procedures which organize descriptive structures. For example, the windows on the wall of a building are recognized by a procedure that specializes in finding regular patterns of rectangular surface markings. This is another example of how procedurally represented information is used in object recognition.

Knowledge about object color and texture — as well as knowledge about the position, shape, and size of an object in the image — controls the interpretation process in Ohta's system. Descriptive pattern-action pairs in the production system determine which actions are appropriate and carry out those actions using procedures, with both the relational pattern-action pairs and the procedures being derived from knowledge about the appearance of the objects in an image.

A system by Nagao [NAG80] also makes use of image-based information. His system is of special interest here because of its use of "characteristic regions." By selecting those regions that can be given a tentative identity (as, say, "a large textured region") with a high degree of certainty and associating a label with those regions ("forest"), the identification process can build up *islands of certainty*, which are initial identifications of possible objects in the image. These serve as a starting point upon which further interpretations are built. It is a technique similar to the "exemplar selection and extension" interpretation strategy discussed in Chapter 3. In order to apply this technique the system requires, in addition to the information about the features of the characteristic regions of the image of an object, information about which of these features will best aid in the recognition of the object. Further, if several related objects are to be recognized, the system can use information telling which objects are typically easier to recognize. Once it possesses this starting basis, the system can selectively invoke specialized procedures for recognizing objects. For example, houses are recognized in three stages. First, the most easily recognized houses are identified based on shape, contrast and the proximity of a dark region (coming from a shadow); these initial house hypotheses establish a context for further recognition of houses. Within this context, a second procedure recognizes houses as image areas that are rectangular and close to previously recognized houses. A final procedure is able to hypothesize houses for which there is only marginal

evidence by noting patches, contrasting in intensity from the background, which occupy the missing positions in patterns of houses.

Nagao's system, like Ohta's, is controlled by a production system, a method of control that is not without its problems. One of the more bothersome of these is the frequent necessity to encode control information by the creation of intermediate hypotheses. For example, in order to produce the final house hypothesis, the three related house processes in Nagao's system communicate through the creation of two intermediate house hypotheses. When the intermediate hypothesis is also one required for further interpretation by the system (as when "house" implies "road") then awkward conventions must be established to differentiate between "the house hypothesis that is a message" and the "house hypothesis that is a true hypothesis." In addition, even when expressing simple procedural tasks (such as, "do A, then B, then C") the discipline of the production system requires additional overhead. Intermediate hypotheses must be used for communication; they are essentially tags, and their use unnecessarily incurs the full overhead of the searching and matching mechanisms of the production system.

Several of the ideas used in Ohta's and Nagao's systems were originally developed in a speech understanding system called HEARSAY [ERM80]. These include the use of a global data structure to record partial results, the use of several levels of abstraction, and global control based on scores from the evaluation of partial results. In HEARSAY the global data structure is called the *blackboard* and is a repository for all hypotheses; hence, it allows the creation of alternative explanations of the data. Within the blackboard, the hypotheses can be associated with various levels of abstraction, each level potentially covering more of the signal. At each level, the hypotheses that are still consistent with a plausible interpretation are selected and grouped by procedures called *knowledge sources*. These groupings

are then represented by a hypothesis at the next higher level of abstraction. Hypotheses deposited in the blackboard can be examined or used by any knowledge source. Each knowledge source, in addition to the creation of a hypothesis, supplies a rating reflecting the likelihood that the hypothesis actually is a correct interpretation of the data. A central control mechanism is responsible for the selection of which knowledge source(s) to activate next based on the ratings of the related hypotheses. Knowledge sources are selected in such a way that those hypotheses which are most reliable are used as a guide to create further hypotheses. Two terms, commonly used when describing this type of mechanism, come from this general "best first" approach to hypothesis expansion: *focus of attention* and *islands of certainty*. Focus of attention refers to the strategies that select hypotheses which are most likely to contribute to the final interpretation; while the term islands of certainty refers to the resulting collections of hypotheses which are relatively more certain (less tentative) than other hypotheses.

In HEARSAY it is also possible to have goal-driven control of the knowledge sources. When missing hypotheses need confirmation (as in completing the most probable phrase), the system creates goals which function as requests for other knowledge sources to create hypotheses for the missing events. The concept of the blackboard provides a general mechanism for communication among cooperating interpretation processes.

One attempt to formalize both the types of interactions needed in interpretation has been made by Tsotsos [TSO84]. In a study of time-varying medical images, he proposed measures for the control of a general search to construct a descriptive network of multi-frame image data. In his system, called ALVEN, a network of object descriptions is organized along four representational dimensions: is-a, similarity, part-of, and temporal precedence. Nodes in the network can be simple recognition

procedures or complex frames with procedural attachment. The organizational dimensions are used to guide the expansion of the interpretation under a hypothesize and test paradigm. He developed several measures of correctness and certainty which are combined and used to select the best hypothesis among alternatives. He also has a measure of conflict and a general heuristic search procedure that attempts to reduce conflict and increases correctness by controlling which hypotheses to test next.

Although the problem that Tsotsos solves is slightly different from that of static image interpretation, his methods may have general applicability. Because he was working in a very restricted domain, it was possible for him to develop a complete detailed network description of his domain and heuristic measures for correctness and confidence in each of the representational dimensions. The general methodology of a structured search using multiple dimensions seems very promising, but its application to more general interpretation problems awaits further research.

Glicksman [GLI82] designed a system for the interpretation of aerial photographs in which cooperating processes are explicitly represented. Objects are modeled as a set of propositions that describe both how the object is expected to appear in the image and what procedures can be used to confirm the existence of the object in the image. Each object model is represented as a schema, a frame data structure with procedural attachment, which includes both a declarative representation of the expected feature values for the object and references to the procedures used for confirmation. These confirmation procedures are used to control the actions for recognizing a particular object. Information about the appearance of objects in the image is expressed in terms of the characteristics of regions and zero-crossing edges. Additional information to guide image labeling is obtained from a sketchmap [MAC78] and from interaction with the user. An overall control program represents knowledge about how these sources of information interact.

In addition to the knowledge about each object the system also represents knowledge about inter-object relations using a semantic network, organized into two hierarchies: "kind-of" and "part-of." A schema is associated with the data through a network of schema instances in which each schema instance is related to a schema by an "instance-of" arc. Also, the schema instances are placed in the compositional and specialization hierarchies.

The combined efforts of the schemas in the system, under the control of a centralized monitor, achieve an interpretation by the construction of a network of schema instances. Schema instances interact through messages which appear in two forms: requests for inclusion in the interpretation of a schema instance in a higher level of the compositional hierarchy, and suggestions for possible interpretations based on context (for example, a road schema instance might make suggestions as to where a car schema is best instantiated). When a schema sends a message to a non-existent schema, a schema instance is created; thus, local control over the ongoing interpretation is implemented by this message-passing mechanism. Global control is implemented through a scheduling list and user interaction. When a message is sent to a schema, that message signals the activation of one or more of the procedures in the schema. The system puts each such schema action on the list. The scheduling list sorts schema actions for execution based on the confidence scores of the related schema instances. The system initiates a cycle, consisting of the execution of one schema program, by selecting from the list the procedure with the largest confidence score. The execution of that procedure might cause other messages to be sent; hence, other programs would be put on the execution list. At the end of each cycle the system presents the user with the opportunity to contribute to the control of the system either through the creation of messages to schema or through the reordering of the list of procedures. In this way, interpretation is

controlled by the procedures of the schemas and by the user through the interaction of sending and receiving messages.

There are three important features to Glicksman's system particularly worth noting. The first is its explicit use of procedural attachment, a direct representation of knowledge by procedures. Some types of knowledge are represented more practically as programs, especially when the knowledge takes the form of "in order to achieve X do Y." In production systems, such as Ohta's system, such knowledge is placed in isolated, atomic chunks. This means that complex procedures must either be represented as "simple predicates" or by a "program" constructed through passing hypotheses from one pattern-action pair to another. In the first case one is led to think of the complex tasks as atomic and not requiring further understanding; whereas, in the second case, the representation of the procedural knowledge is spread out and care must be taken to avoid unanticipated interactions. In Glicksman's system, however, the procedurally represented knowledge is centrally placed — in the schemas. Each schema is understood to represent a set of complex actions which are self contained. A second important feature is the use of schemas to represent control information. Messages between schemas implement the greater part of the control in this system and this control knowledge is encoded in the schema procedures. The third important feature results from the inclusion of the user in the cycle of interpretation. Having the freedom to rearrange the priorities of the system, the user can redirect interpretations, causing the system to selectively explore particular hypotheses or avoid the continuation of misguided interpretations.

2.3 The VISIONS System

Many of the ideas and methods developed in this dissertation have their intellectual roots in the research into an architectural framework described by Hanson and Riseman [HAN78]. While a full description is given as the topic of the opening

sections of Chapter 2, three key ideas are introduced here as central to our image interpretation system: the use of a representation of intermediate levels of abstraction which relates image features to object models, the application of the hypothesis and test paradigm, and the use of object schemas to represent procedural information.

Objects in the VISIONS system are represented at many levels of abstraction. The original description separated those levels into three groups. The first group, in which regions, segments and vertices are represented, is closest to the image. In the implementation for this dissertation much of the information at this level of abstraction is represented as special patterns of lines, corners, and image features for special purpose matching routines; for example, see the first roof interpretation strategy in Chapter 3. A sustained effort has gone into developing methods for extracting regions and segments from the image to match the image level of abstraction. Nagin [NAG79] and Kohler [KOH83] developed region segmentation algorithms, while more recently Burns [BUR84] developed methods for extracting lines from the image data.

The second group of abstractions consists essentially of three-dimensional primitives: surfaces and volumes. Within this framework, York [YOR81] examined ways of relating two-dimensional abstractions based on general curves to three-dimensional abstractions, although his effort was never integrated into the full experimental system. As with the first level, the current implementation uses this level of abstraction in special purpose routines for constructing object descriptions.

The last level of abstraction is the object (and scene) level where schemas reside. The implementation of that level of abstraction is the topic of this dissertation. Having the image-based features which match the two-dimensional level of abstraction is only the first step. The next problem is matching those features to the correct

abstractions in a description in such a way as to eventually hypothesize the correct object.

The general paradigm of hypothesize and test has been used for matching the various levels of abstraction to descriptions grounded in image data. This is the method investigated by this dissertation. The representation of objects and the control of object recognition routines are combined in schemas which describe object (or scene) classes. The concept of a schema as a mechanism to control computer vision was introduced by Hanson [HAN78] and refined by Parma [PAR80]. The core idea is that specialized knowledge whose general application would be too expensive can be applied to advantage when a specific hypothesis is being tested. For example, a routine that finds pairs of parallel lines, when run on all the lines in the image, will produce many meaningless pairs; when there is a hypothesis for a particular rectangle, however, the routine can search for parallel (or nearly) parallel lines in a restricted portion of the image. In a similar framework, Reynolds et. al. have applied inferencing interpretation mechanisms to the problem of multiresolution image labeling [REY84].

In an attempt to make the image interpretation process more general and robust, Lowrance [LOW82] developed a general method for inferencing over relational graphs based upon evidential support distributed over a set of hypotheses. This method has been applied by Wesley to the problem of selecting objects based on discriminating features [WES82] and to the control problem of action selection [WES83]. Early experiments, using a simulation of the interpretation process, indicate the potential usefulness of these inferencing mechanisms for controlling computer vision [WES86].

This dissertation extends many of the ideas developed by the research associated with the VISIONS project. We have highlighted the key ideas of multiple levels of

abstraction, hypothesis and test, and the schema because they are themes of this dissertation.

2.4 Knowledge for Interpretation

We now turn to the questions that opened this section and summarize what we have learned from the work of others. Quite naturally, we will highlight those lessons relevant to understanding the system described in the subsequent chapters. In addition, we will draw parallels between the design of our system and that of others in order to highlight differences. The opening questions were: What types of knowledge are used by interpretation systems? How is the knowledge represented? What method of control is used?

In response to the first question, several types of knowledge are used by interpretation systems, with some based on knowledge about the three-dimensional geometry of objects, while others rely primarily on knowledge about image-based measures and the two-dimensional characteristics of possible projections. The systems that deal with complex images require knowledge about many types of image features and the ranges of values typically associated with them. Examples of a few of these are the average color and texture of a region; the expected size and position of a region; and the length, orientation, and contrast of an edge. Any of these features, singly or in combination, might indicate that a particular interpretation is appropriate for some portion of the image.

In addition to knowledge about the properties of objects, we have seen that knowledge about interobject relations is useful. The most prevalent relation is that of part-to-whole, forming a composition hierarchy. Other inter-object relations include attachment, co-occurrence, and spatial relations. The spatial relations are sometimes expressed as ranges of values in either two dimensions (as, "sky is above ground") or in three dimensions (as, "the angle between the plane of face a and that of face b"). The systems that we reviewed also represent knowledge about how to

select and group image features. In at least one of these systems (Glicksman's) there is a clear representation of the relation between possible objects and the processes that group image features to make an interpretation of a portion of the image. In addition, Tsotsos [TSO84] proposes that interobject relations can be used to organize search space for hypotheses and to test for conflict among hypotheses in a systematic way. While it may eventually be possible to use interobject relations in a general way, our system relies on object specific knowledge when controlling the direction the search should take.

Our model of an object consists of both types of information. In our system we use a limited class of knowledge about both the three-dimensional geometry of the object (where this will aid the interpretation process in a straightforward way), as well as knowledge about the expected appearance of the object. While our system uses three-dimensional information, it is not as systematically described as in Brooks's system. Our purposes differ from Brooks's. Whereas he set out to show how to use a general three-dimensional model, we are concentrating on other issues and use three-dimensional information as part of other processes. One of our goals is to examine which types of knowledge and forms of control would be useful in interpreting outdoor scenes. Also, we are interested in studying how to coordinate different types of information. Although we use a simple type of geometric model (based on polygonal patches), the associated matching procedures illustrate the utility of combining geometric information with other types of information. Like Glicksman, we have included information about the type of process that can best be used to select image features and to group both features and hypotheses to form additional hypotheses.

Turning to the second question, "How is this knowledge represented?", two representations which we want to highlight are the semantic network for representing inter-object relations, and the schema (or frame with procedural attachment) for

representing the knowledge about an object. When working with complex objects we must rely on the use of multiple features and inter-object relations to overcome the intrinsic ambiguity of the data, the errors introduced by the feature extraction processes, and the inherent ambiguity of the imaging process. Each of the systems reviewed represented some knowledge procedurally. However, only with the use of schemas are procedures explicitly represented, such as in Glicksman's system and in ours; in all the other systems the references to procedures are treated as symbols or as predicates that test for the existence of symbols. Parma [PAR80] describes how schemas were intended to fit into the VISIONS framework. Like Glicksman, we use schemas to represent the knowledge associated with an object, and part of that representation refers to the interpretation strategies, which are procedures for object recognition.

The full description of the structure of our schemas is given in Chapter 2, but a quick preview will contrast what we do with Glicksman's use of schema. The key difference lies in our treatment of the relation between the schema and the image data. In Glicksman's system a schema is associated with image data via a data structure called an instance which has parameter values some of which refer to the image features; in other words, the instance represents the interpretation of the image data. In contrast, our system presents the instance as a locus of interpretation activity. The instance produces a hypothesis structure representing a partial interpretation. In our system, the hypothesis (as opposed to the instance associated with the schema) represents the interpretation of the image data. We were influenced by the style of the knowledge sources of HEARSAY; however, rather than associating knowledge sources with a particular type of signal or a particular level of abstraction, in our system knowledge sources are related to scenes, objects, or parts of objects. By separating the activity of a schema from the hypothesis for the existence of the object, we avoid the problem typical of production systems,

that of tentative hypotheses being put on the blackboard too early. In our system, a schema instance can delay the creation of a hypothesis while it invokes procedures to select and group features, thereby increasing confidence in the possible existence of the object. In this sense, each schema is a self-contained knowledge system; it may carry out any number of internal "hypothesis and test" cycles before "announcing" the hypothesis on the blackboard.

In response to the last question, we examined methods of control. However, as this issue is central to the rest of the dissertation, we will devote a separate section to it.

2.5 Discussion of Alternate Control Methods

The problem of control is important in any system that has to deal with large problems or large solution spaces. Many factors must be considered when deciding which actions to take in image interpretation, i.e., which image operations to apply, which grouping processes to invoke, which subroutines to call, which hypotheses to consider, which goals to pursue, and which contexts to establish. Many of these decisions are guided by the current global context of the interpretation, and one of the more difficult problems in control is the combination of local evidence to establish global context. In this section we will briefly contrast six methods of control, some of which were used by the systems reviewed earlier: relaxation labeling, inference network confidence score propagation, special purpose procedures, production (or rule-based) systems, algebraic constraint manipulation, and schemas.

In relaxation labeling local evidence is encoded into likelihood scores for local labelings and then a globally consistent set of local labelings is determined by iteratively updating the likelihood scores to be locally more consistent. Relaxation labeling is the method of control used in the system by Barrow and Tenenbaum [BAR76]. Some of the advantages of this method are that relational information can be directly expressed as labeling constraints and that there is a uniform method

of propagating information from local labelings to global constraints. Also, it is a natural method to use when information is available in terms of confidence in particular labels. Two problems with this method make it difficult to use in complex settings. First, it requires an estimate of compatibility coefficients to be used by the update rule; these coefficients encode the relational information between potential labels at neighboring locations. In complex environments, compatibility can depend on several factors and it is frequently better to be able to examine those factors separately. The interpretation of complex images relies on multiple interacting sources of knowledge. These sources deal with entities at differing levels of abstraction with differing complexity of description. Further, the cost of using this knowledge and invoking processes to infer structure can vary greatly, requiring a greater flexibility of control than is offered by traditional relaxation labeling. While there are obviously interpretation problems for which one might use relaxation labeling, we believe it is not a sufficiently rich control method for interpretation of complex scenes. The schema-based control in our interpretation system has the facility for integrating fine-grained knowledge from diverse sources.

Another, perhaps deeper, problem with relaxation labeling is the use of confidences or probabilities to estimate the validity of a label. Using a single number in this way confounds evidence for the object with information about the reliability of the source of the evidence and information about the likelihood of processing error. In addition, since the compatibility coefficients are used to express an estimate of the effect of correct, errorless evidence for one label on the correctness of using a particular neighboring label, the probability of truth and the estimates of error are further confounded by being combined with the labeling scores from neighboring locations. In interpretation, there are really three questions that need to be answered: How much evidence do I have to lead me to believe that this is the correct label? If I use this label as the correct label, how much evidence do I have that it is free

of error? and What relative weight and combination method do I use to take into account the contribution of evidence from multiple sources? While it is clear that these questions are related, it is not clear that using the weighted average of one number per label adequately captures the interactions between error, information, and semantic relations. Unfortunately, our system does no better on this issue. Specialized procedures, called interpretation strategies, are used to determine the likelihood that an object exists, and while some of the more completely designed interpretation strategies do return information in a scoring vector which records error and certainty estimates, methods for combining and using these methods have been left for future work.

Some attempts have been made to overcome the inherent problems of using probabilities for estimating confidences. In MYCIN [SHO76] and PROSPECTOR [DUD78] likelihood scores were used to reflect the confidence or belief in propositional rules of the form:

if (list of antecedent clauses in conjunction) then (consequence clauses).

Since a network can be formed based on the appearance of a clause in the consequence in one rule and in the list of antecedents of another, they used heuristic rules (derived from probability theory) to propagate the effect of scores related directly to evidence through the network of clauses, assigning scores to all the clauses. In MYCIN, Shortliffe tried to overcome some of the problems of combining scores by associating a *certainty factor* with the rule which was used in combining scores.

Additional attempts to develop formalisms for the propagation of evidence in inference networks have led to methods of dealing with information that is uncertain, incomplete, and inaccurate. Such information is called evidential information [LOW82] and has been applied as a means for making control decisions in computer

vision [WES82, WES86]. A formalism for expressing this type of information, introduced by Shafer [SHA76], provides a more flexible mechanism for separating the two factors of confidence in a label and ignorance as to the correctness of the label, as well as formal mechanisms for combining evidence. This formalism also provides measures corresponding to the notions of inconsistency and plausibility. In a relational structure, the confidence measures can be combined to propagate the uncertainty information in a manner analogous to reasoning. As with relaxation labeling, using an inference network provides a general uniform method for control, but it is not clear that it can also be flexible enough to handle the diverse types of processing used in interpretation. The use of an inference network requires that measurements from diverse sources be reduced to a common scale and combined using general methods that do not reflect specific knowledge about the relations among the sources. Although the propagation of evidential measurements over an inference network is a theoretically attractive alternative, is not clear that these methods for pooling of evidence are of pragmatic utility in large systems. We did not employ an evidential reasoning approach in our research, but it appears that it would function best as *a part* of a larger system.

Special purpose procedures have also been used as a means of control in computer vision. The systems by Roberts [ROB64] and Paul [PAU77] rely on specialized programs to guide matching and the construction of an object description. As more specialized procedures are added to a system, the relations between them can become more complex; thus, the more general the system, the more it might need to rely on additional control. The success of these specific procedures, however, suggests that there is a place for them within a general control framework. Specifically, such procedures can be selectively applied when a plausible context has been established. Further, specialized procedures are also a part of general control strategies.

Used for measurement and conversion of numeric quantities to symbolic labels, specialized procedures occur in many of the propositions in production systems, in the measurements in constraint propagation systems, and in matching procedures in both these types of systems. The schema-based system used in this research relies heavily on specialized procedures: these are the interpretation strategies associated with the schemas. They contain procedural information for controlling the interpretation process within local contexts.

An interesting approach to this is the method of algebraic constraint manipulation. Where relations can be cast as algebraic expressions and evidence combined as limits on parametric values, then techniques for symbolic reasoning using algebraic expressions can be applied to determine the implications of the evidence. Thus, a known set of parameter value ranges can be used to predict possible ranges of unknown parameter values. Brooks [BRO81] combines this method of algebraic constraint manipulation with a global strategy of recognizing major parts of an object before the detailed subparts. In each stage of his process, matching image features to possible objects produces a set of constraints which implies possible configurations of object descriptions and, thus, limits which possible matches should be considered in the next stage. This use of algebraic constraints provides a means for local information to have global consequence. It also provides a means for diverse sources of information to interact. Unfortunately, it requires a mechanism for reasoning about algebraic expressions. Further, when the constraints can not all be expressed algebraically, then a method is needed to convert symbolically expressed constraints to algebraically expressed constraints. Even when all the constraints can be algebraically expressed, it is not always possible to manipulate them to get well-behaved constraint relations (see discussion in [BRO81]). The effectiveness of this method of combining information awaits further study.

A production system using a global blackboard addresses the problem of how to control multiple interacting sources of information. This method was introduced in the HEARSAY speech understanding system [ERM80]. In contrast to methods using relaxation, inference networks or constraint manipulation, in which evidence is encoded primarily as numerical confidence scores or through parameter values, a production system emphasizes the symbolic expression of evidence. The vision systems of Ohta [OHT80] and of Nagao [NAG80] use this method of control. By expressing the results of each operation symbolically and the antecedent to each operation as a pattern of acceptable symbolic expressions, applicable actions can be selected by pattern matching on the symbolic expressions. The results of those actions, expressed symbolically, can be used to select new actions. Each production, or rule, of a production system represents a relation between entities in the interpretation model. For example, one such rule might be, "If a region is bright blue, large, and high in the image, then label it tentative-sky." Each such rule can be thought of as a separate statement about the relations between entities. Ideally, the user "programs" a production system by encoding all the relevant relations in a set of rules and the production system proceeds with interpretation. Each rule is an independent statement of fact and they interact only when the facts are related.

In practice, there are two major related problems with using production systems. The first is that the "scope" of the atomic relations is quite limited and the second problem is that explicit ordering of operations is antithetical to the principles of production systems and the style of their use. In a production system the "grainsize" or scope of the typical rule has some undesirable effects. A rule typically represents only one relation among objects; thus, complex relations, especially those that are context dependent, require the introduction of additional symbols and artificial intermediate labels. This restriction is related to the problem

of the naturalness of expressing complex control sequences. While production systems permit a natural and modular expression of simple relational rules, the more complex the control sequence the less natural the expression. Not all control information is naturally expressed in if-then style relations. Much control information is most naturally expressed as programs or program-like statements: "do step A while incrementing n and collecting set S until condition B occurs; then for each element in S do step C." While such specifications of sequences of actions and conditional invocation can be expressed as symbolic rules, it is not a natural method of expressing them. Further, while production systems invite an incremental construction of knowledge the development of methods for systematically adding rules to such rule-based systems is still under study.

Image interpretation relies on fine-grained knowledge with complex interrelations. Consider the "simple" problem of recognizing a window in an outdoor scene. Much of the image area of the window might look like other things: reflections of sky, curtains hanging on the inside, or reflections of nearby foliage. Often there will be only partial image evidence for the geometric forms and continuations of the building surface that will suggest that this is a window. The recognition of the window requires knowledge about the structure of buildings, the existence of other objects in the image, and the effect of the reflectance properties of glass. To write general rules about these properties, such that they would always be applicable, would require that the rules be complex. The point is that general monolithic control systems, while attractive in a theoretical argument, do not provide the flexibility needed to integrate fine-grained knowledge, especially when the understanding of that knowledge is evolving with the development of the system. It would be better to have some way of relating knowledge directly to the task of establishing a particular goal; then the knowledge can be expressed in the form of procedures specialized to the given goal.

In the systems that we reviewed here, the control procedures can be distinguished by the degree to which they are centralized. In Roberts's system, as well as in the one developed by Barrow and Tenenbaum, one central control program operated with a single description of objects and produced a single result. This contrasts with the move away from centralized control, first suggested by the HEARSAY system, and later applied in Ohta's and Nagao's image interpretation systems. The distribution of control, however, is preceded by the distribution of knowledge. Although these systems use a decentralized representation of knowledge, and that knowledge includes some control knowledge, they still depend on a single, central mechanism to select which procedure to activate next.* Later systems combine the distribution of control knowledge with a centralized (but minimal) monitor. For example, in Glicksman's system the central controller is a scheduler and most (if not all) of the information about what action to pursue next comes from the interaction of the schema instances and their programs. We based our system design on a similar decentralization of control. An advantage of distributing control is that adding new elements is easier. Glicksman makes this point when discussing the inclusion of the user as a source of information. In his system the user is treated as another element of the system, passing messages to schema in order to control them.

The problems common to the methods of control discussed above stem from the tentative and explorative nature of our understanding of image interpretation. Because of the experimental nature of image understanding systems, general methods of control do not serve us well. While relaxation labeling, inference propagation over a network, production systems, and algebraic constraint manipulation are all attractive ways of managing information, each has flaws along with the promise. The necessities for using clusters of fine-grained local knowledge, for having to describe

* Also, [LES77] and [COR81] describe experiments to distribute control in HEARSAY.

both processes and structures, and for wanting to experimentally and incrementally build up our knowledge mitigate against using such general control structures. Another problem is the need for mechanisms to utilize and propagate information about error and uncertainty. Finally, there is the need to use contextual clues to limit the computation of complex grouping or evaluation functions to those areas of the image where they apply.

As the number of objects and the complexity of the types of knowledge increase, interpretation systems rely increasingly on control guided by heuristics. Each of the systems reviewed here (except for that of Roberts and Barrow and Tenenbaum) used a pattern of control that can be characterized as "find the best first." When dealing in a domain such as scene interpretation, where most matches between the data and the object model are ambiguous and tenuous, the search for possible matches can be limited by starting with the best matches and selecting subsequent matches by following the relations among objects. This is the concept of "focus of attention" which we introduced when discussing HEARSAY. Much of the guidance for matching comes from the richness of relations among objects, especially among attached parts of objects.

Many of the problems with the previous approaches can be overcome by using schemas (or frame-based knowledge structures). Each schema is a collection of information about an object, including the relations between parts of the object, a description of the geometric structure of the object, and strategies for recognition of the object. In our selection of schemas for control we believe we have found a method that provides the flexibility for experimental development while allowing the clustering of both descriptive and control information. Moreover, while our system does not manage to propagate uncertainty information, we attempt to contain its effects by selectively controlling the paths of interpretation based on the establishment of contextual information. Schemas can be used for data-directed

or goal-directed control; they provide a means for combining special purpose procedures with relational information; they facilitate the accumulation of supporting information for the hypothesis of an object to reduce uncertainty; and, finally, they cluster information about an object, effectively creating object-centered knowledge sources which represent (at a larger grainsize) the object and associated local control.

3. AI and Machine Vision: Schema and Frame

The idea of unifying information in a structure associated with a major concept is not new. It has been discussed in the psychological literature for some time. Piaget's early work (cf. Furth [FUR81]) used the term *schème*, which was mistranslated as schema (more currently translated as scheme) and adapted by English-speaking psychologists. This term was used to label an adaptive assemblage of sensory-motor correspondences that were organized in association with an event class. Piaget postulated such structures to account for the observation that perceptions of similar events changed qualitatively over time. His explanation for this observed change was that the internal structure representing the primitives for perception and their interactions changed. Piaget's later work and the interpretations of it differentiate between scheme and an abstraction of an assemblage of schemes that is amenable to symbolic manipulation. This latter concept is more closely allied with the present usage of the term schema.

Similar considerations were expressed by Bartlett [BAR32]. In his study of memory and its functioning, he proposed a version of schema to account for his observations of the systematic patterns of recall in his subjects. Bartlett defines the term "schema" as an organized active structure for the grouping of sensory information:

'Schema' refers to an active organization of past reactions, or of past experiences, which must always be supposed to be operating in any well-adapted organic response. That is, whenever there is any order or regularity of behavior, a particular response is possible only because it is related to other similar responses which have been serially organized, yet which operate, not simply as individual members coming one after another, but as a unitary mass. Determination by schemata is the most fundamental of all ways in which we can be influenced by reactions and experiences which occurred some time in the past. All incoming impulses of a certain kind, or mode, go together to build up an active, organized setting...

(page 201)

He goes on to stress the role of schema as an active structure:

Remembering is not the re-excitation of innumerable fixed, lifeless and fragmentary traces. It is an imaginative reconstruction, or construction, built out of the relation of our attitude towards a whole active mass of organized past reactions or experience, and to a little outstanding detail which commonly appears in image or in language form.

(page 213)

Although we are not concerned with modeling human memory, the ideas developed in this chapter were derived from these concepts of schema. We are especially interested in the idea of a schema as a structure describing past events in a way useful for recognition of future events, a structure that has active components or processes which *impose organization* on incoming sensory events. From this brief review of the general concept of schema, we have seen how memory is thought to require a combination of active and descriptive elements.

These general concepts of schema have been applied to the study of human vision. Biederman performed a set of studies (see [BIE73], [BIE81], [BIE82], and [BIE83]) dealing directly with human vision and his work led to a model of perception which includes the concept of schema. By studying the effects of image organization on perception he attempts to discriminate between the effects of recognition

based on data and local organization ("bottom-up" processing) and schema-driven image recognition ("top-down" processing). In a typical study [BIE81] he describes an experiment in which he shows subjects images of natural scenes for a short interval (150 msec) and asks them to confirm the existence of a target object in a scene. Biederman observes that the objects more frequently identified correctly are those consistent with expectations based on the type of scene; and, further, that the types of errors made by subjects suggested that they are making hypotheses of object identity consistent with the type of scene shown. The effect of establishing a context by schema activation is observed in a short time (within one fixation, according to Biederman). Further, the strategy of using that context appears to be so strong that even when the best strategy would be to ignore the context, the subjects still make errors which can be explained by assuming that their judgments are based on an assumed context.

The idea that people perform interpretation by first recognizing a scene type and then using that context to guide recognition is also borne out in an experiment performed by Akin [AKI80]. In this study he showed subjects pictures of different resolution and asked them to describe the scene. Errors made by the subjects suggested that they recognized details based on the assumptions derived from the context of the scene, an effect that became more pronounced as the detail became less perceivable (at lower resolution).

From these experiments in human perception there are several principles we would like to emphasize as motivating the use of schemas in this dissertation. Information in memory is organized: there are relational links, hierarchies of detail and generality, and so on. The unit of organization, a schema, has internal structure and associated processes which are used to impose organization on sensory events and construct descriptions of data. These constructed descriptions are the "recognition" (or recall) of things "known to the organism." In summary, the

schema is an active structure that selects and organizes sensory events to construct interpretations of "what is in the world."

Arbib has developed a similar notion of schema in [ARB78], where he talks of active assemblages that organize sensory events to direct action. This effect may propagate through a sequence of schema, with the "actions" of one schema being the "events" organized by other schemas. He proposes that schemas are hierarchically organized, with schemas of larger scope influencing rough, approximate, overall behavior and schemas of lesser scope supplying detailed responses. Overton has refined this concept of a schema [OVE84] for application to the domain of robot control. In his system a schema consists of three major portions: an activation-section, an event-section, and a memory-section. The activation-section contains predicates (possibly procedures) used to determine when and how the schema should be active. The event-section contains the programs that gather and group sensory data and create the actions associated with the schema. Overton suggests that there are situations where the programs in the event-section would need to operate in parallel. The memory-section represents the adaptive component of the schema and contains structured traces of past activations that can be used to modify the schema to make it more effective for the task(s) being performed by the programs in the event-section.

Related to the concept of schemas, the concept of frames has a similar history. Goffman, in an analysis of expository discourse [GOF74], suggests that frames express contexts for understanding. He uses the term frame in a fashion similar to our use of the term schema, as an organized assemblage of past events.

Minsky [MIN75] defines a frame as a parameterized prototype for a concept. He proposes that the description of an object be a collection of related frames, each of which describes constraints on the possible interpretations of perceptual events.

The effect of these constraints could be limited, tuned, or otherwise controlled by supplying values for parameters that "fill slots" in the frame.

In this dissertation we use the term schema because we want to suggest and emphasize the active nature of structures for interpretation and recognition. We have found useful the theory that memory contains not only an abstraction of past experiences, but also abstractions of perceptual processes.

4. Summary

For effective computer vision the representation of objects should include many types of information. The description of three-dimensional geometry alone is not sufficient. In order to recognize objects in natural scenes, we need to use information about the color and texture of the image regions in the projection of the object. In addition, we need to use relational information between object parts and between objects. When relations can be represented so that they are expressed both in terms of the scene (three dimensions) and in terms of possible images (two dimensions) then the constraints derived from such relations aid in the task of interpretation.

Having representations of the objects is not enough; we must also be able to use those representations efficiently. This is the problem of control. We have reviewed many ways to effectively control image interpretation: production systems, network representation with central control, distributed knowledge sources acting on a central representation - to list a few. The major problem to overcome is the coordination of the numerous, diverse types of processes that are required to make effective use of the many types of information that are potentially useful.

The choices of the type of representation and of the method of control are intertwined. Each style and implementation of control affects the way in which information is represented. Thus, for example, we have seen that using a production system - while it permits a distribution of information - requires a propositional

style of process description. The use of frames (or schemas) facilitates the combining of declarative descriptions with the procedures that manipulate them.

Choosing a control strategy for computer vision seems to be dictated by the need to combine declarative and procedural information. Much of the knowledge needed for image interpretation is symbolic in nature: relations between objects, color and texture labels, object part placement, and object labels; but many of the symbols are derived by processes that are essentially numeric (e.g., statistical measures, mechanisms for determining variable thresholds, and feature extraction). More importantly, given our current understanding, the strategies for when and how to extract symbolic information are more easily and naturally expressed as programs. This requires that the processes employed to initiate and drive the use of the symbolic information be an integral part of the object representation. In doing so, we must assure that the system can work equally well with the declarative and procedural information. In the pages that follow we will describe our network representation which includes procedural attachment with distributed control.

CHAPTER II

System Design and Architecture

In this chapter, we present the basic design of the schema system and place it in the context of the VISIONS image understanding system. The design addresses two issues: representation of the knowledge required for interpretation and mechanisms for controlling the application of knowledge to the interpretation task. We have elected to combine both the descriptive information and the control information into an "object centered" structure, called a *schema*. The collection of schemas necessary for an interpretation task are combined into a *schema network* which captures the relations among individual objects (e.g., a house schema) and expected contexts (e.g., a house scene schema). Further, we define *interpretation* as the process of building a network of nodes which represents the objects in a scene, their relations to one another, and their rough spatial position. This network also contains primitive nodes relating image data directly to object parts. In the experiments described in this dissertation the network is constructed from the evidence available in a single digitized photograph of the scene.

Each schema describes both an object to be recognized and the methods for recognizing it. These methods are expressed by *interpretation strategies*, programs referred to by the schemas, which control interpretation by directing attention to image events, by selecting object models for consideration, by deciding which object relations to examine, and by recording partial interpretations for future reference.

The interpretation strategies rely on object models; thus, a schema also includes the description of the object. The selection of appropriate actions is based on the relations between evidence in the image and information about the object. Partial interpretations tell the system what has been understood so far and the object

(and scene) models in the schemas predict which image events to expect. Such interactions can help overcome interpretation problems introduced by missing detail (resulting, for example, from occlusion or lack of resolution for distant objects), failures of the data abstraction processes, or failures of interpretation routines.

The schema network is part of the VISIONS image understanding system. In Section 1, we begin with an overview of the system, concentrating on the portions of it that support interpretation. Section 2 is an outline of the schema structure, Section 3 illustrates how objects are described, and Section 4 discusses the mechanisms for communication and control that are available to the interpretation processes within a schema. The details of each interpretation strategy will be presented in Chapter 3.

1. An Overview of The Interpretation System

In designing this system, our goal was to satisfy three general requirements. First, we allow for the exploration of the types of knowledge needed to recognize objects (and interpret scenes). The system design permits the inclusion of previously developed programs for object recognition. All of the knowledge associated with an object resides in a *schema* for that object, thereby achieving program modularity in the knowledge engineering process. The schema represents both declarative knowledge such as propositions about inter-object and object part relations, and procedural knowledge such as programs for object recognition. Second, during the interpretation of an image the system produces a description of the scene, called an *interpretation network*, which is represented as a semantic network. Third, partial interpretations that do not interact can proceed in parallel. Since the possible relations among objects and scenes are represented in the schema network, interpretation strategies in the schemas can determine when schemas can be activated in parallel.

1.1 Background - The VISIONS System

Our interpretation system is embedded in a machine-vision system that has been developed over a ten-year period by the VISIONS research group at the University of Massachusetts [HAN83]. In this section we briefly describe its architectural framework with an emphasis on the placement of the interpretation system within that framework.

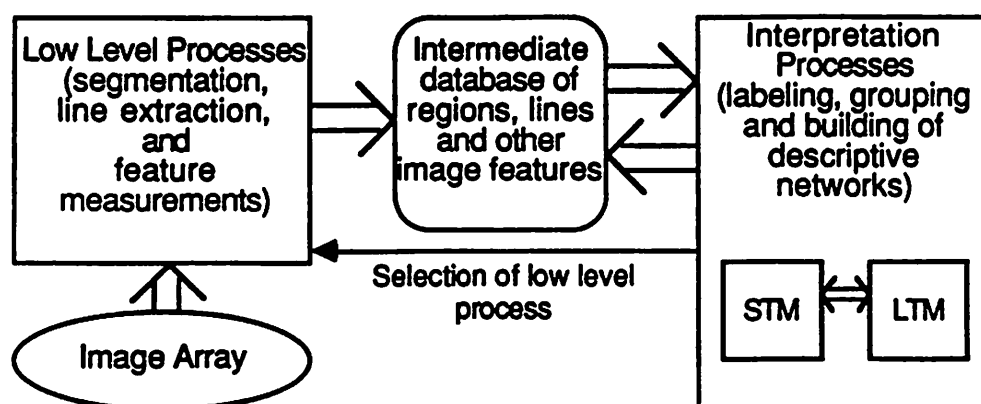


Figure 5. Overview of VISIONS System Architecture

Initially, abstractions of image data are made by processes working independently of knowledge about particular scenes or objects. These abstractions permit the initiation of the interpretation process. Partial interpretations, based on early hypotheses, guide the application of specific knowledge about scenes, objects, and their relations to build more complete interpretations. These interpretations can, in turn, guide the further abstraction and reorganization of image data.

For our purposes the problem of machine vision is divided into two cooperating processes (see Figure 5). The first process is that of data abstraction. A set of routines is available, each of which can be applied to the image to make an abstraction of the input data, either a segmentation and some measurements within the segmented regions (e.g., average intensity) or a description in terms of significant lines and their features. These data abstraction procedures construct the base level representations of the image data.

The second process is interpretation, in which intermediate data abstractions are associated with the base level representation through symbolic structures that describe the events in the scene. A hypothesize and test strategy is used to construct the scene description. Initial partial interpretations are derived using inexpensive labeling strategies or from the context established by prior partial interpretations. An object label is then associated with the partial interpretation and a schema for that object (or scene) type activated. Once activated, the schema provides procedures to direct further construction of the partial interpretation by the creation of intermediate abstractions that are in closer agreement with the stored symbolic description associated with the schema. As shown in Figure 6, the various levels of abstraction interact.

The exact nature of the interaction among stages in the interpretation system depends on the types of information available. When there is little information about the scene or objects in it, especially during the beginning phases of an interpretation, these two components operate in a data-driven fashion: the initial phase of the interpretation depends on the descriptions produced by the segmentation processes. However, there are times when the type of scene or object being interpreted is known; for example, it is either given before interpretation, or derived from an earlier partial interpretation. In these cases the interpretation process can control the segmentation and data abstraction processes. Thus, we have two phases of communication. In the first, the segmentation does not rely on knowledge about a particular scene and information flows from the data to the interpretation processes. Once interpretation has started, however, the interpretation process has both partial interpretations and intermediate data abstractions to work with. Segmentation can be selectively applied and the construction of new data abstractions can be controlled by programs that group previous regions, lines, and hypotheses. New interpretations evolve based on the intermediate data.

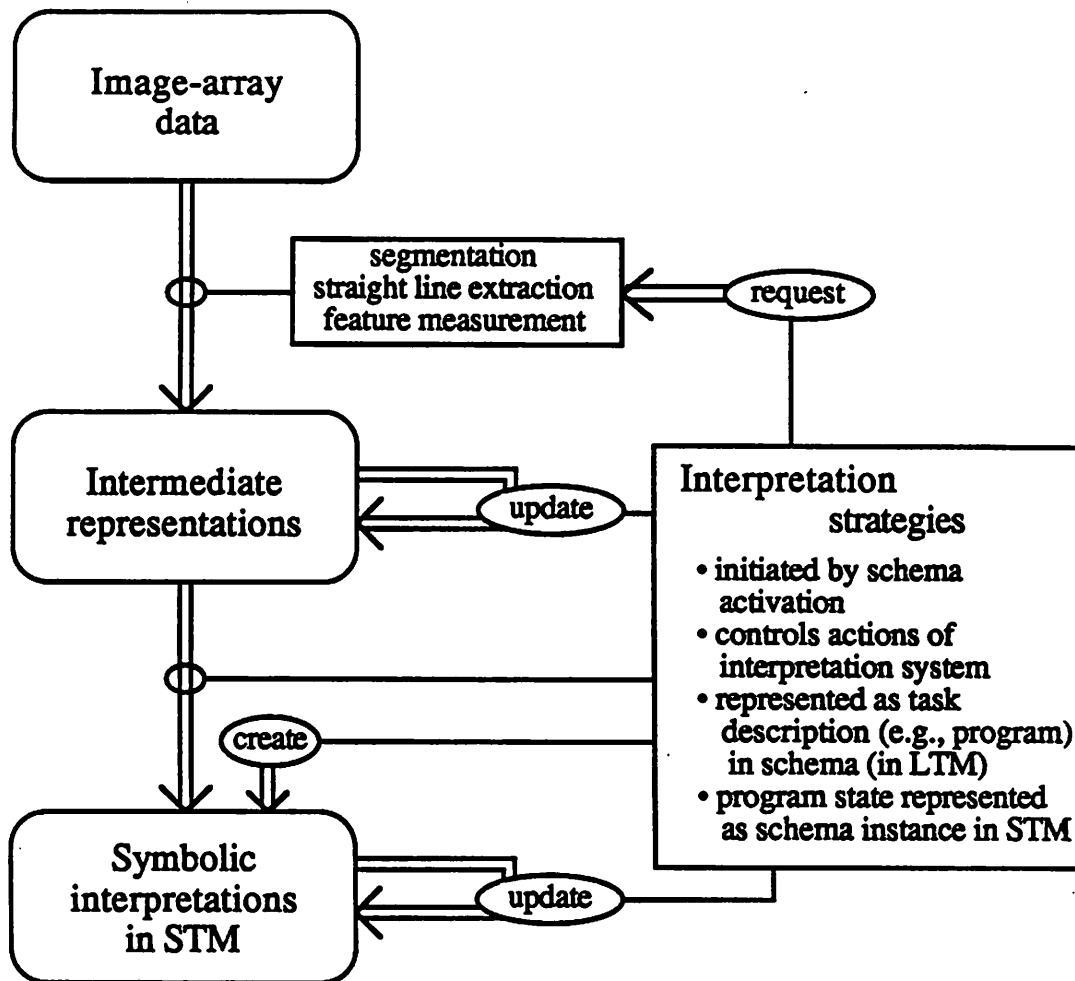


Figure 6. Multiple Levels of Representation and Processing in VISIONS

The transformation from the numeric image array to the symbolic interpretation is mediated by intermediate representations and controlled by interpretation strategies. The intermediate representations include many forms that can be interpreted both numerically and symbolically. For example, the description of a region contains the average values for its component colors. This is used in numeric comparisons between regions, as well as an identification of region color for processes that label the region. The interpretation strategies control the creation of abstractions within these intermediate representations.

In the VISIONS system, the interpretation process mediates between two data structures, Long Term Memory and Short Term Memory (see Figure 7). Long Term Memory (LTM) is the collection of all the object representations and control

Figure 7. Representation of Short and Long Term Memory

The schema, in LTM, contains several levels of abstraction. Scene schemas describe the collections of objects and their likely arrangement in space, while object schemas describe the relations among object parts. Relations among the objects and scenes are represented in a relational network. Some object schema are primitive in that they describe no further parts. Further, much of the knowledge relating objects to possible images of the object is implicitly represented in interpretation strategies within the schema. Thus, the parts of the network under "object description" depicted in the "details of object schema" as a network in LTM are incorporated into interpretation strategies. Using the LTM structure as a guide, the strategies control the construction of corresponding networks in STM which associate object nodes with image data. This is indicated by the brackets on the networks and the line from LTM to STM connecting them.

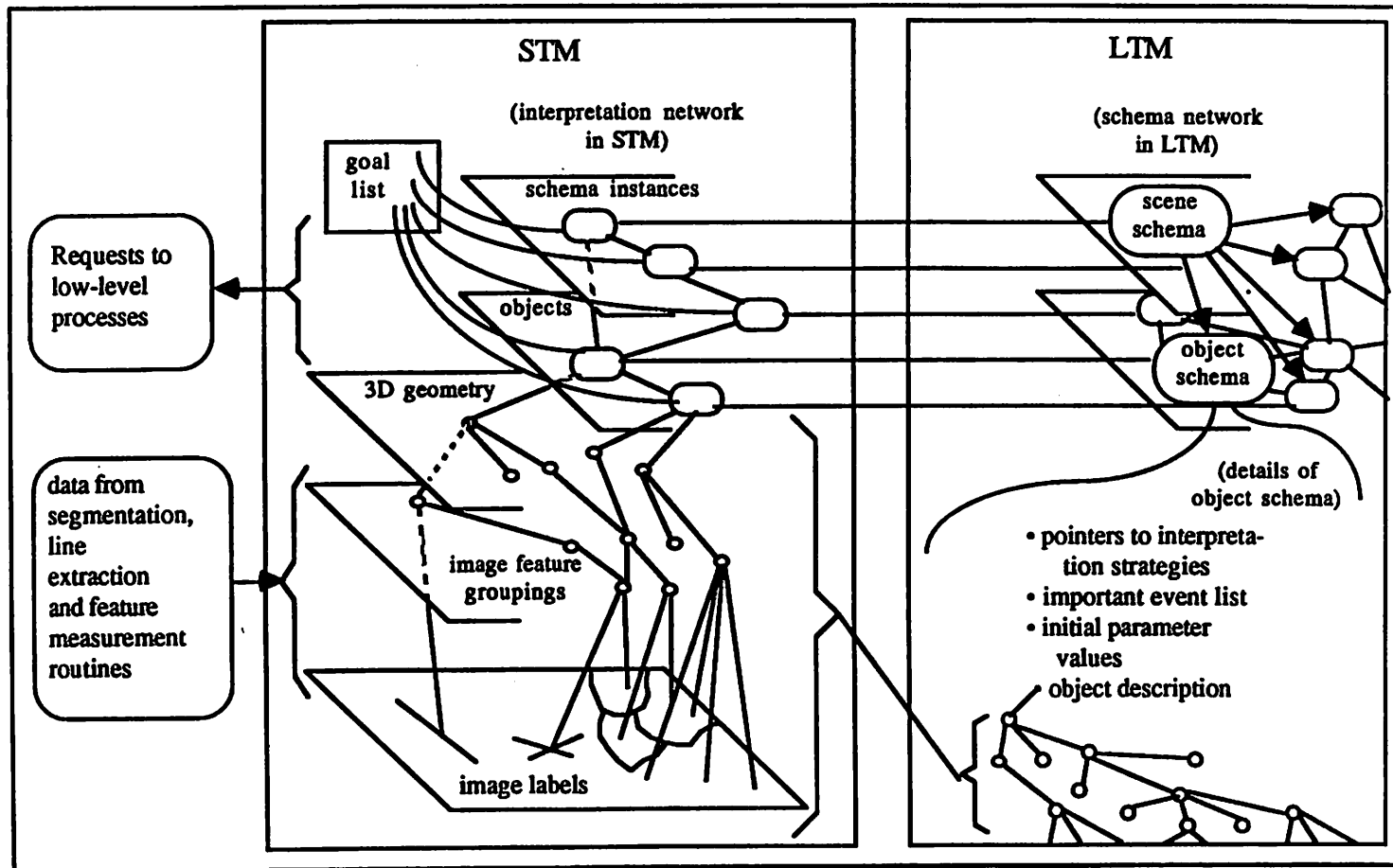


Figure 7.

information. For our interpretation system the information in LTM is organized in a network structure, consisting of *nodes* connected by *arcs*. The nodes describe scenes, objects, and object parts that the system can recognize.* The arcs represent the relations among those objects: symbolic spatial relations, geometric spatial relations, and hierarchical relations.

Schemas are nodes in the LTM network (schema network) which describe objects and scenes. Both the methods for recognizing the object and the description of the object are contained in the schema. The recognition methods are procedures called *interpretation strategies* which are executed by the interpretation system. The object description serves as a prototype for the creation of the descriptions of the object in Short Term Memory (STM) during image interpretation.

While LTM contains the general description of objects and scenes, STM holds the description of a particular scene and the objects in it. The application of the interpretation strategies from schemas activated during interpretation causes the creation of object descriptions in STM. Relations among objects link these descriptions into networks representing partial interpretations of the image. The networks in STM have some nodes associated directly with data from the image and others which represent groupings of those items into hypotheses. When the processing is finished, the networks remaining in STM represent the final interpretation.

1.2 The Images and Segmentation Routines

We can better understand what is involved in the construction of a description if we examine the input used by our system and the output that it is expected to

* We use the term "object" in a very general sense, including sky, grass, foliage, the ground-plane, roads, sidewalks, and other similar things which are not commonly thought of as objects. Further, in the discussion of schemas, a schema can represent either a scene, an object, or an object part. However, for the sake of brevity we frequently refer to schemas as a representation of an object and leave to the reader's understanding that scene and object part are implied.

produce. In this section we describe the type of input received by our system. Figure 8 illustrates two images with their associated regions and lines; these pictures of suburban outdoor scenes exemplify the type of image our system is able to interpret. Our interpretation system interacts with the segmentation systems which produce the data abstractions shown in Figure 8. Additional input is supplied by feature measurement routines.

A description of the image input, segmentation, and feature measurement follows. Each of the images used for interpretation was first digitized from 35mm slides using a 50 micron spot size on an Optronics flying-spot scanner. They were digitized in three colors (red, green, and blue) at 8 bits resolution per color, with the resulting digitized images being 512 pixels on each side. In order to make computation reasonable for the segmentation algorithms, the images were averaged using non-overlapping windows, to a resolution of 256 pixels on a side. In this averaging process, four pixel values in the original image contributed to one pixel value in the average image; each color was averaged separately and the results were truncated to 8 bits. The results of the data abstractions are shown in Figure 8; the region segmentations were produced by a region merging algorithm [GRI85] and the line extractions were based on a segmentation of the gradient direction [BUR84].

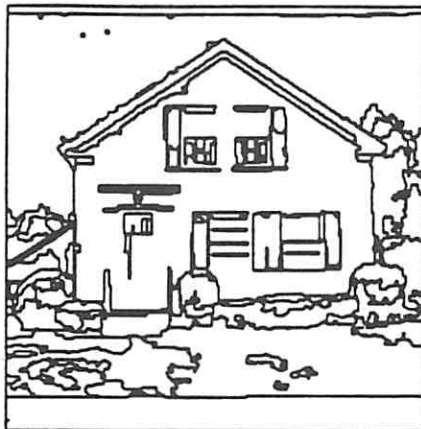
In the region-based segmentation and feature extraction [GRI85], a database is built which contains attribute measurements for regions in the image that are roughly homogeneous in their color or texture. The underlying motivation is the expectation that large areas of the image which are homogeneous often correspond to large portions of an object surface. The region segmentation procedure starts with an oversegmented image and merges regions using a best-first merging strategy guided by a heuristic merging rule. The initial oversegmentation is a simple region-growing algorithm with a very constrained threshold, which results in only a few large regions, with most of the image covered in small regions (under five pixels).

Figure 8. Data Abstraction Processes

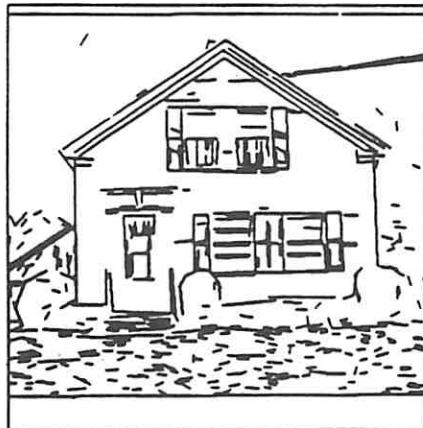
This figure shows example images together with the regions segmented and the lines extracted from the image data. (a) Two images from the experimental database; (b) their region segmentations and (c) the set of long, high contrast straight lines. These image-based data are used to initiate interpretation.



(a)



(b)



(c)

Figure 8.

The region merging algorithm tests each boundary by scoring the two adjacent regions as to how suitable they are for merging, and merges those regions found most suitable. The merge score is based primarily on color and texture similarity, relative region size, and length of the boundary. The selection and merging continues until the minimum score is above a threshold. The algorithm keeps track of scores, only updating those that change from a merge, and has a method of finding the best merge based on sorting the merge scores which considerably speeds up the selection and merging iteration. Associated with this region segmentation is a feature extraction routine, in which measurements of color, texture, shape, size and location are made for each region. These, along with the region identity, are part of the base level of the intermediate representation.

The straight line extraction, described in [BUR84], is based on the idea that gradients in the image intensity which lie along a connected straight line are often caused by edges in the scene. Like the region-based method described above, this abstraction consists of segmentation followed by feature measurement. The segmentation, based on gradient direction, is achieved in a two-step process. First, the image is segmented into regions having the same general direction of intensity gradient over the region. Then, a line associated with each gradient region is determined by intersecting a plane fit to the intensity surface of the region with the plane of constant average intensity value. Finally, features are extracted from the gradient regions and their associated lines. These features include the length and position of the line, the slope of the gradient plane, and the average intensity of the region. They, along with the description of the line, are part of the base level of the intermediate descriptions.

These two abstractions, regions and straight lines, form the basis of the subsequent interpretation, and the representation of knowledge within the system must take into account the type of image information available. For example, the roof

schema contains information about plausible roof surface orientations which can be matched to information about global surface orientation from the data. In our case, however, it is first necessary to construct some intermediate representation, because we use no method for obtaining local surface orientation directly from the image. Thus, the roof schema (in LTM) needs to have a representation of the knowledge required to derive surface orientation from the regions and lines available. The representation and use of these types of knowledge are discussed in Chapter 3.

1.3 Objects Used in Interpretation

In addition to the data abstraction methods, the design of the interpretation process is also dependent on which objects are to be represented in its Long Term Memory (LTM). Since the interpretation produced by our system is based on object identification, we examined a variety of house scenes and developed a list of objects that the system should be able to recognize (see Figure 9). The objects are shown in a summary version of the schema network. (The full network is shown at the beginning of Chapter 4.) In addition to the objects, an interpretation system would eventually have to distinguish among various scenes. However, in order to keep the task of knowledge engineering to manageable proportions, we include only one type of scene, suburban house scene, with its superclass, outdoor scene. Both the input data and the classes of objects described delimit the system's interpretation capabilities.

1.4 Image Interpretation and Expected Interpretation Results

In any task that requires knowledge, the goals of the task will determine the representation of knowledge and the content of that representation. Because we wanted to develop a general system, our goal was to have it label the image and build a description of the major visible objects in the scene. Thus, we required our system to produce a semantic network which describes those objects (as nodes) and gives their relations (as arcs).

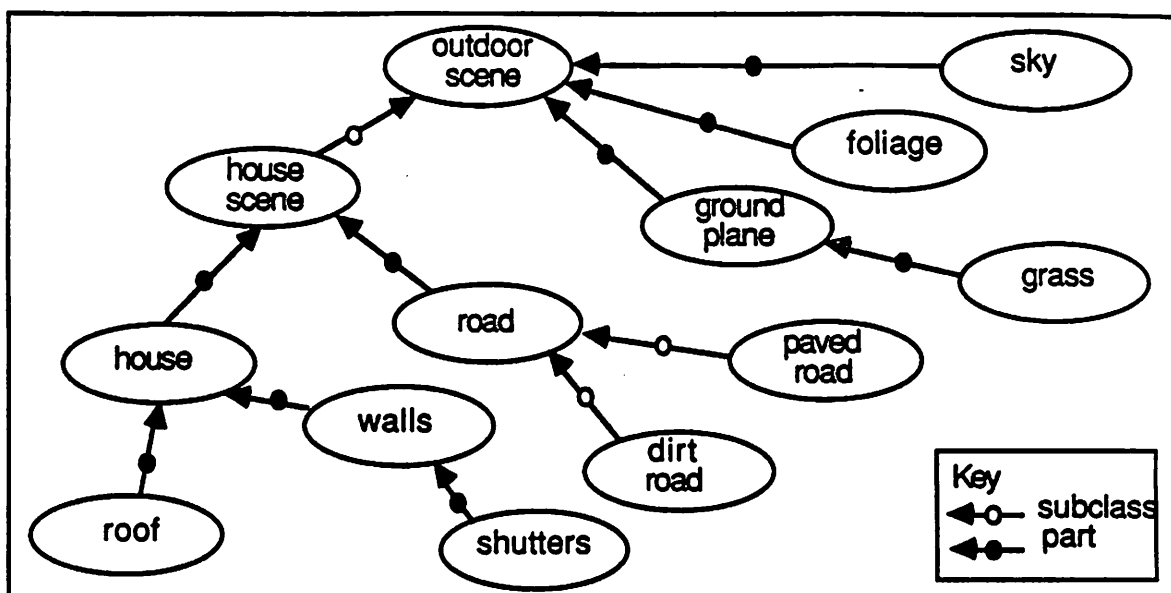


Figure 9. Object Classes for Interpretation

The objects used in the schema network are shown with their compositional (or "part-of") and specialization (or "subclass") relations. These are the schemas in the (LTM) network used by the system in interpretation. Many more objects and object parts are represented in the information implicit in the interpretation strategies. For example, the walls schema, during interpretation, generates nodes for each house wall. See the example of the roof schema, described in the text, and Figure 13.

As a specific example of the type of output we expect our system to produce, Figure 10 shows such a network constructed by hand by the author.* Given the photograph shown in Figure 1 (Chapter 1) this is the style of results that would be expected: a network of object instance nodes (sky, grass, etc.) and their relations (sky above house, window on wall, etc.). In addition to the type of network illustrated here, the system can produce numbers associated with each node that represent confidence values. They are an estimate of the confidence associated with

* Contrast this with the network shown in Chapter 4, produced by the system.

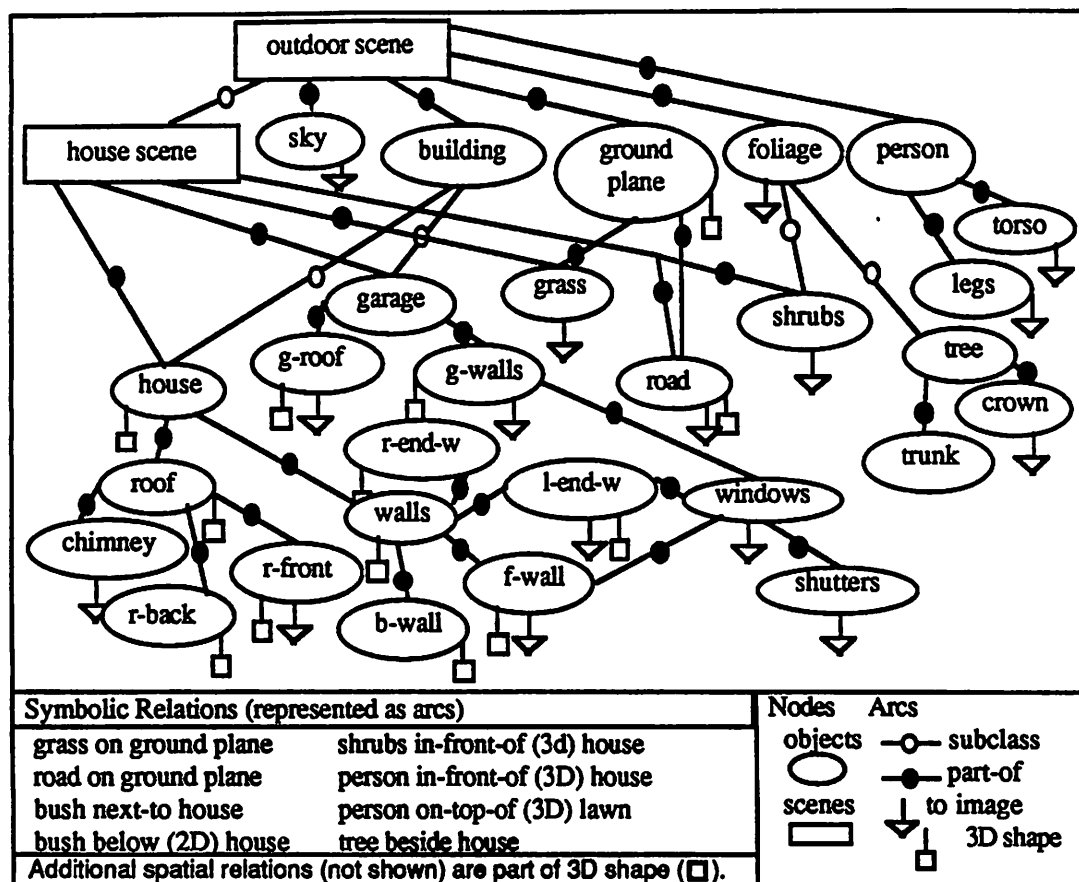


Figure 10. Hand-Generated Example of an Interpretation Network
 A possible interpretation network for the picture of Figure 1, Chapter 1, is shown. For clarity, much of the structure is not explicitly illustrated. The symbolic relations listed under the graph would actually be represented as arcs in the network. Moreover, each reference to image data and to three-dimensional descriptions (shown in the graph as a triangle or a square, respectively) would also be a set of nodes and arcs describing those additional entities and relations.

the assertion that the presence of the object is supported by actual visual events (data, data groupings and abstractions, and other intermediate symbolic elements).

Although we have no specific application in mind, several tasks can be imagined: an English language description of the scene (see [CON82]), a query and response

system, robot planning and navigation, or an image interpretation executive system (requesting interpretations of particular images). We believe that the type of semantic network produced by our system is a reasonable representation for any subsequent task requiring a description of the image.

2. Schemas as a Representation of Objects

The general problem of interpretation is one of matching a model of an object or scene to groupings of the data abstracted from the image. However, there are many possible objects and scenes, many possible groupings of image features, and many possible correspondences. In addition, the process of comparison for even one correspondence can be computationally expensive. Therefore, there is a need to limit the number of matches attempted.

Some portions of the model are more easily matched to the image than others. From these early matches, we construct a skeletal description of the image. That description, in turn, guides further matching. For example, several methods of grouping data and ordering the recognition of subparts might be equally useful in recognizing the house. Consider the roof as it relates to the house. Recognition of the roof creates an initial hypothesis for the roof, which can be used as a partial interpretation by the processes associated with the recognition of the house. Other relational information can be used to achieve the same end. The interpretation of the house might depend on isolating the sky ("house below sky"), on delineating the ground-plane ("house on ground-plane"), or upon the recognition of other subparts (shutters, for example). Each such method describes a different strategy for recognition and these related strategies form part of the information about the house.

In our system, the schema contains both the strategies for the recognition of an object (or a scene) and the information applied by those interpretation strategies.

The interpretation strategies draw upon descriptions of the geometric form of the object, representations of relations among objects, and indications of the views of the object. In addition, the interpretation strategy is a program which "describes" the process for recognition. Thus the information for controlling the interpretation is partitioned by its usefulness to the recognition of a particular object class.

There are benefits that accrue from the partitioning of the knowledge by associating a schema with an object. The schema provides the potential for parallel processing of the image. Each different type of object is associated with a separate process description. Since objects are spatially separated in the image, this partitioning makes it possible for the recognition processes for such separate objects to be activated at the same time. With the addition of a representation of the *activity* of the process (i.e., a program state), multiple instances of the process can be simultaneously active. Thus, it is also possible to have the same schema working on several portions of the image, creating the descriptions for multiple instances of the object.

Another benefit of a representation based on schemas is that the development of knowledge can proceed separately on each object. The schema is a modular representation of knowledge. The processes for control and organization of information and the description of the geometry and image characteristics of the object are all contained in one independent structure. This allows the development of new object descriptions in an incremental fashion.

3. Declarative Representation of an Object

Each schema contains a description of the object. This description can include a description of the geometric form of the object, or it may contain references to the image features of the object. Much of this information is represented directly as values in a declarative frame-like structure. This declarative representation has the

advantage that it can be directly manipulated by the interpretation strategies but is still (reasonably) independent of them. Also, in as much as declarative representations are similar and common across several schemas, more general interpretation strategies can be used. Ideally we would like to be able to represent all the features of an object in a declarative fashion. Pragmatically, however, the recognition of some object features is dependent on complex processes where the sequence of processing steps determines part of the information necessary to recognize the object. A procedural representation is used for information of this type. In this section we describe the declarative portion of the schema, which we call an object description. To motivate the discussion we provide an example: the description of a roof.

3.1 An Example Representation: House Roof

We have chosen to represent one of the many possible types of roofs. A standard dictionary lists five types of roof: Lean-to, Gable, Hip, Gambrel, and Mansard [RAN75]. The roof type that we want to describe is Gable (see Figure 11), consisting of two surfaces, both rectangles, which meet at a common line (at the crest of the

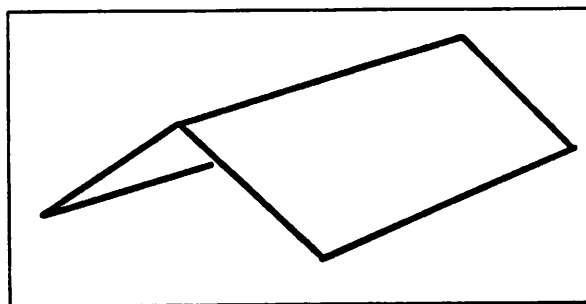


Figure 11. Gable Roof

The spatial relations of the gable roof model are represented as geometric properties of the rectangular surfaces of the two faces of the roof. Other attributes are attached to the schema as ranges on parameter values and are implicitly represented in object specific interpretation strategies.

roof). Because we wish to describe the roof as it might appear on a typical house, the angle between the two surfaces is restricted to being between 45 and 105 degrees. Since the color and texture of the two surfaces of the roof are the same, the image areas corresponding to the projections of the surfaces of the two rectangles are expected to have the same color and texture values. Because the color and texture are empirical values, developed from image data, they are represented as functions of image feature values.

The pictorial depiction of each view of the roof (see Figure 12) illustrates how views can be characterized by the groupings of lines and the relations among them. The side view (Figure 12a) is characterized by the fact that the top and bottom edge are nearly horizontal (when the direction "up" is known), and they converge to a distant vanishing point; in addition the side edges taper in as they go up due to perspective distortion. The end view is characterized by the upward pointing peak (Figure 12b) and the combination view (Figure 12c) is characterized by the set of features common to the other two views. Key features can be identified from an examination of these views and are presented in the description to distinguish among the views (Figure 12d). Features such as these augment a description of the three-dimensional geometry to distinguish among views.

The part of an interpretation network of a house-scene which represents a roof is shown in Figure 13. Here we see the relations as they are expressed both in the schema as part of LTM and in the interpretation network in STM. Note that while some of the description of the roof is directly represented as slots in the schema (for example, the limits on the angle between the faces of the roof are supplied as limits associated with the slot pitch-angle), other information is represented as part of the interpretation strategy (for example, the relations among the lines that form a rectangle).

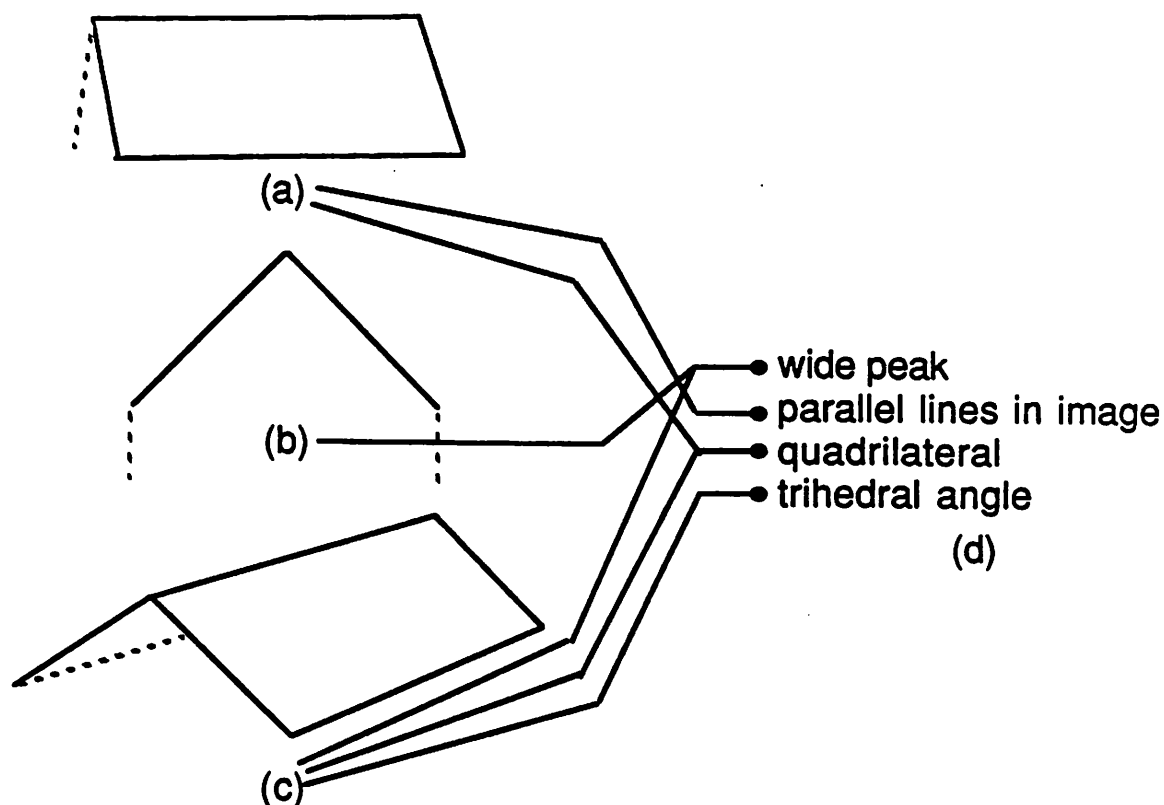


Figure 12. Views of the Roof

Part of the information in the schema and its interpretation strategies is a representation of possible views and their relations. This figure shows three "typical" views of the roof and the type of image features that would help identify each view. (a) A side view; (b) an end-on view; and (c) a more general view. (d) A list of major features for the roof is related to each of the views. For example, both views (b) and (c) are considered when a "wide peak" is detected, as shown by the lines from that attribute to those two views.

In Figure 13 we show the roof schema from the schema network in LTM and a corresponding schema instance in STM (both shown as boxes). The other nodes shown are those constructed by the activity of a roof schema instance during interpretation (in STM). Both faces of the roof are associated with instances of a rectangular surface. The instance node has four associated edges identified in counterclockwise order (when facing the outside surface of the rectangle). A special

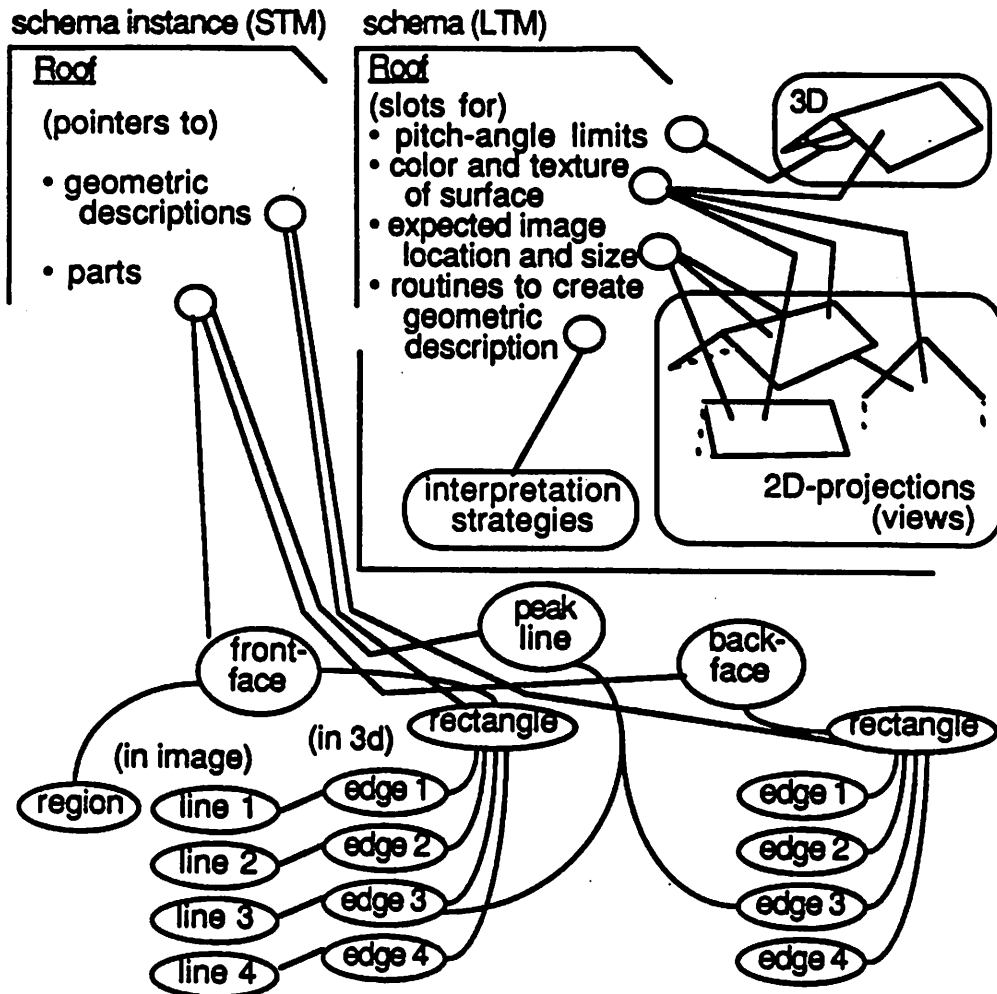


Figure 13. Network Describing Roof

This is a sample of the type of network generated for an object by the schema interpretation strategies during the interpretation process. The interpretation network shown here, the STM portion, is taken from the interpretation network produced in the first example discussed in Chapter 4. It contains references to both the image data (shown as lines and a region) and the derived three-dimensional structure (shown as edges and rectangles). In addition the nodes have the actual parametric values of object descriptions (e.g., lengths of lines, their common endpoints, and the angle between the roof faces). This roof description is actually part of a larger network (not shown here) that includes additional information. For example, the common edges between the house walls and house roof are also part of the network. The interpretation network is derived from the information in the schema, in LTM, which contains information on the spatial relations of the object parts, indicated here by the sketch of the roof (labeled 3D). In addition, image-based information is contained in "views," shown here as sketches of the 2D-projections.

edge in each rectangle is identified as the common edge at the peak of the roof by an association with the object part, called peak line. In addition, the front face of the roof (defined as the visible face) is linked to image regions and (through the instance node) to the image lines corresponding to its edges. Finally, color and texture information are represented as the features of the associated region.

There is a question as to whether each part of the roof (e.g., the roof-edges) should also be represented as a schema. In this example, since the parts of the roof are matched by the interpretation strategy for the roof (see the roof interpretation strategies in Chapter 3), there is no real need to make separate schemas for each edge. The description of the geometry for the roof is part of the roof schema. However, it is easy to imagine a situation where having separate schemas for each part is a reasonable approach (the house interpretation strategy, in Chapter 3, illustrates this). Whether LTM contains a description of the part as a separate schema, as a descriptive structure associated with the schema, or as an entity to be generated by the interpretation strategy associated with the schema, depends on the generality of the interpretation strategy and the ease with which a general description can be matched to the image data. Thus, while it is true that all schemas are nodes in LTM and the schema related nodes in STM have corresponding LTM nodes, not all the nodes created in an interpretation network (in STM) have corresponding LTM schema nodes.

3.2 Three-Dimensional Geometric Structure

In this section and the following two we discuss, in turn, three topics related to the description of objects: the representation of object geometry, the representation of expected image feature values, and the representation of the relations among and between these two types of information. The description of object geometry is accomplished through a polygon-facet representation. We chose not to use a more complex representation of three-dimensional shape because we wanted to avoid the

difficult problems of matching three-dimensional models to the two-dimensional image data. Instead we have chosen to concentrate on how to use two-dimensional representations in matching (with the addition of three-dimensional geometric information when possible).

Our object geometry representation is based on line segments, polygons, and polyhedra, the constituent components of a polygon-patch representation. For example, each face of the roof is described as a rectangle represented by a node with an associated ordered set of four lines in space; the roof is described as two joined rectangles. Additional information is necessary. For example, to describe a square, we need to be able to represent the facts that all the line segments have equal length and that the angle between adjacent line segments in the polygon is 90 degrees. The parametric information is captured using slots on the nodes representing the geometric entities. These slots are filled with values or parameters which provide any constraints needed.

Additional information is expressed in the procedures for recognizing the constituent polygons. For example, the description of the four sides of the roof's rectangular faces is embedded in the procedures for recognizing rectangles which identify the image data and build the description of rectangle. All of the detailed geometric entities are represented by procedures for the construction of polygons which are invoked by the interpretation strategy in LTM (Long Term Memory). When an interpretation strategy recognizes an object that has a particular type of geometric structure, it calls a procedure to construct the corresponding geometric description in STM. The constructed description has the relations specified explicitly.

3.3 Image Features

The object description used in our system provides structures for integrating representations of three-dimensional geometry and image-based features. In the example of the roof, we saw that the roof can be described in terms of a few typical

views (Figure 12). In any such view, additional relations among expected image features can be expressed. For the view of the roof shown in Figure 12a the surface appears in the image as a quadrilateral with two sets of nearly parallel sides. One set of lines is nearly horizontal. Such facts as these can be used to determine the possible views of the object. When matched, a view is associated with a particular set of image features by the creation of a description of an instance of the view. Moreover, the view has a pointer to the description of the related object. Thus, the information represented in the view can be used to advantage. The matched view is used to select which part of the object description will guide the construction of the network that specifies the object instance.

There are cases where a description of two-dimensional image-based information, expressed as a view, constitutes the representation of an object. Consider a tree, for example; unlike the roof, the geometric structure of this object is complex and not easily represented. While there is geometric structure, we are without a method for economically representing it or methods that effectively use such representations. Instead, we choose to represent the tree by summarizing the features of the image of the tree from a typical viewpoint. Some of the notable characteristics of a tree image area (e.g., a set of regions) are that the boundary of the area is "jagged" and that the area can be expected to have texture and color values within certain ranges. Further, with the assumptions about camera position and orientation implied by using a view, facts about the tree's expected size, orientation, and position in the world can be expressed as limits on the position, shape, and size of its image regions. Features such as these characterize the tree.

The inclusion of image feature information in the object representation has proven very useful. For example the features "green" and "textured" are sufficient to identify a few key areas in the image likely to be grass. We have developed a rule-based labeling function (described in Chapter 3) that uses a description of the

range of feature values characteristic of regions belonging to the typical image of the objects in order to locate instances of those objects in the image. This description of feature ranges is part of the declarative object description.

3.4 Relations

The relations between schemas form the arcs of the schema network. There are several types of relations. All of the object descriptions are embedded in two hierarchies: the composition hierarchy and the specialization hierarchy. These are expressed by the relations, represented as arcs between nodes, of "part" and "subclass," respectively. In addition to the two hierarchies, there are spatial relations which are only used in the two-dimensional views of objects and in the interpretation network. The particular relations, always given with respect to the frame of reference of the camera, are: near, above, to-the-left-of, and to-the-right-of. There are also two symbolic relations that occur only in scene descriptions and have to do with assemblages of objects. They are "attached-to" and "occurring-with."

3.5 Summary

There is a tradeoff between a procedural representation of the object geometry and the approach using a descriptive structure and a general matching routine. By encoding the *process* that is to be used to gather data, group primitive events, and match the geometric structure to the image data, many levels of interaction are brought under control of a single monitor, thus avoiding combinatoric explosion. For example, the line that corresponds to an edge is usually fragmented requiring that the line fragments from the image be grouped. The blind application of a line fragment grouping routine to the image would result in a lot of unnecessary work. Further, the search for the four lines (groups of fragments) that correspond to the edges of a rectangle involves a larger space. However, knowing that the rectangle is part of a roof means that the interpretation strategy starts with a fixed location in the image and some information about which line (or line fragments) should be part of the description.

The line fragments that the interpretation strategy is first able to find determine the type and order of search for other lines. That set of line fragments also determines which model lines are to be sought. Many of these process-related decisions are best represented as procedures.

On the other hand, a declarative representation of the parts of the roof and their relations allows a greater flexibility and the application of more general strategies (for example, "try to confirm all unmatched parts"). In situations where additional information can be used to control the combinatorics, then the more general declarative representation would be desirable.

Whether the current implementation has the right mix of declarative representation and procedural representation is a question. It is clear to the author that in some cases it does not. A system that relied more on an explicit representation of the intermediate levels of representation would be more flexible. Ideally the information about the objects would be separated from the interpretation strategies and encoded as descriptions of the object which could be used to control the search in a more general way.

4. Combining Declarative and Procedural Knowledge

Schemas control the interpretation process through the activities of a *schema instance* that is initiated when a schema is *activated*. The schema instance stores the state of the interpretation strategies associated with particular image events. Once activated, the schema instance causes the construction of a description of an object or scene from the image. During an interpretation, several schema instances will be active at the same time. Some of these instances* need to communicate

* We will refer to a schema instance as an *instance* when such reference is unambiguous. They could also be referred to as "schema instantiations" or as "the state of an active schema."

results and requests with other instances; the communication techniques developed are a subset of those commonly used in object-oriented languages [GOL83].

Each schema is represented as a data structure with four parts. In addition to the object description covered in the previous section, a schema has three structures related to control. The first is a list of important events: these are the types of hypotheses or image data abstractions which will cause the activation of the schema. They can be thought of as a simplified set of patterns in the style of production systems and are discussed below and in further detail under event-driven activation (in Section 4.1). The second structure is a set of parameter values each of which represents the initial, default value for a local variable in the schema instance. These are similar to the slots and values of frame data structures. Finally there is the list of interpretation strategies associated with the object.

Particular data events indicate that a schema should be active. The "important event list" is a list of the names of those events. Actually it is a list of the names of hypothesis types which correspond to those events. When an event is detected, a hypothesis for that event is created, signaling the schema that a schema instance should be created. The resulting instance interprets a part of the image, usually that portion of the image associated with the original event.

The set of parameter values defined as part of the schema serves as default parameter settings and is the initial setting of the local variables in the schema instance. In the discussion of schema instances (Section 4.2) we will describe how these parameters are used as local variables with initial values. Particularly in the case of goal-directed activation, these local variables are used to parameterize the schema at activation and to differentiate among instances of the same schema during interpretation.

The list of interpretation strategies, the third control-related structure of the schema, indicates which programs can be used to create and manipulate hypotheses for the object. The interpretation strategies make the correspondences between data and the hypothesis for the object; that is, they construct object hypotheses, create additional hypotheses where appropriate (e.g., geometric descriptions) and add relational arcs to existing partial interpretations in STM. (See Chapter 3 for additional details of the interpretation strategies.)

The interpretation strategies are written in a Lisp-like programming language developed specifically to permit simulation of parallel execution. Specifically, the language contains two pairs of wait-and-post control statements: one for communication between interpretation strategies within the same schema instance, and the other for communication between instances (see Section 4.3). In each of these control statement pairs, a wait statement causes the suspension of the execution associated with the issuing interpretation strategy until the corresponding event is posted. If the corresponding event were posted prior to the wait, the issuing interpretation strategy would continue without pause. In the case of communication between schema instances, a message is returned from the call to the wait command.

4.1 Activation

There are two ways in which a schema becomes activated: goal-driven activation and event-driven activation. The chain of occurrences in the first case commences when a goal is created by an interpretation strategy of an active schema instance. For example, if a house-scene schema needs part of the image interpreted as a house, then it makes a request for that interpretation by issuing a goal for house. The goal is given a set of parameters that are to be "passed on" to the schema instance responding to the goal. These are used to convey specific information about the object requested through the goal, i.e., ranges of color or expected image area. A schema that potentially can satisfy that goal gets activated and is associated with

the goal (see Figure 14). A goal-type-to-schema table determines which schema should be activated in response to a goal.* This activation causes the creation of a schema instance (discussed shortly), which includes the invocation of one or more interpretation strategies to satisfy the goal. When the goal request results in the activation of a schema the resulting schema instance "claims" the goal. The identity of the goal record is returned to the requesting instance and can be used to access the control flags associated with the link between the goal and the instance which claimed the goal. The goal name can also be used to send requests to the satisfying instance (as described shortly).

At some point in the execution of an interpretation strategy, if there is sufficient support, a hypothesis is created and sent to the requesting instance as a hypothesis to satisfy the goal. (See the discussion of hypothesis creation and verification in Chapter 3.) The requesting schema instance determines if the hypothesis has been created by testing flags associated with a goal description which was returned when the goal request was made. The requesting instance may wait for the hypothesis to be returned by issuing a wait-for-response command. This command suspends the execution of the requesting instance until the responding instance sends a message to the goal: either a hypothesis that satisfies the goal or a report that no such hypothesis can be created. When the responding instance either reports failure or returns the hypothesis, the flag in the goal description is set to indicate the presence of the hypothesis causing the execution of the waiting instance to be resumed with the associated message returned from the call to the wait-for-response command. If either the message with the appropriate hypothesis or a "no hypothesis" message is sent prior to the time that the requesting instance issues a wait command, then the

* Generally, there might be more than one type of schema that should respond to a goal request but this raises the issue of deciding which schema instance actually works on satisfying the goal; that is, which schema instance "gets the contract." Thus, in this system, we have only one schema type per goal type.

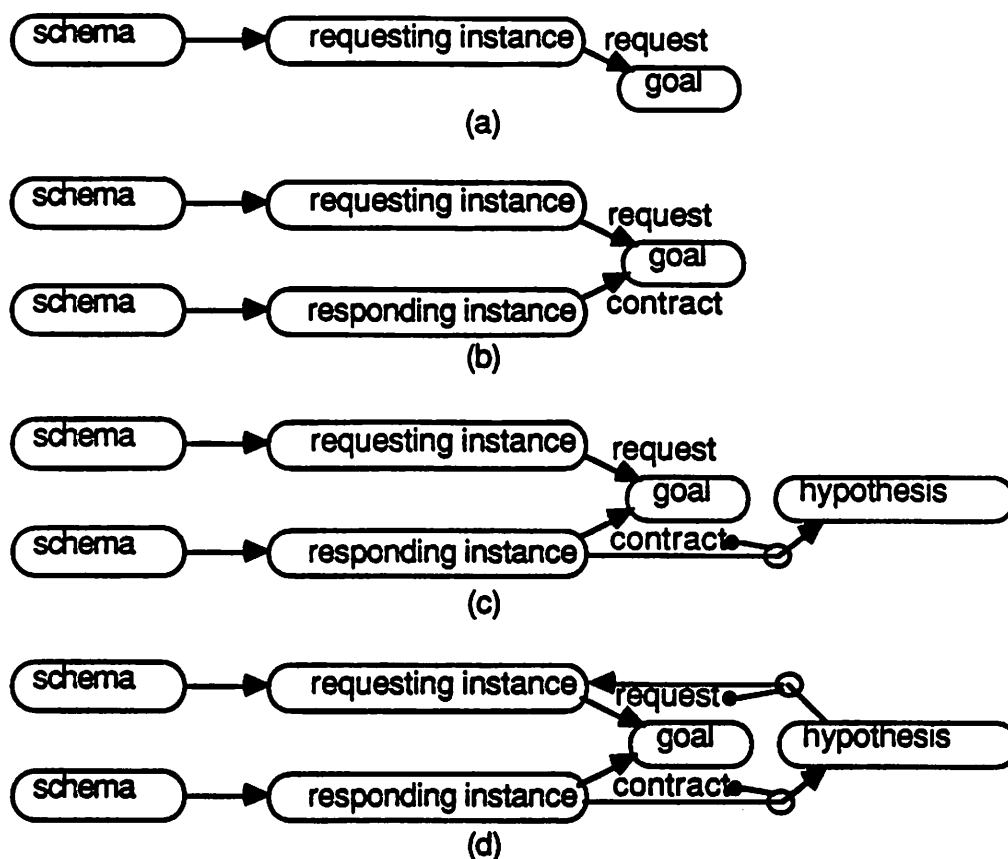


Figure 14. Relation of Schema Instance to a Goal

When a schema becomes activated to satisfy a goal, the schema instance thus created is associated with the goal. (a) During interpretation the interpretation strategy associated with a schema instance requests a goal. A request link holds the information associated with the request. (b) This goal causes the activation of a schema which will attempt to satisfy the goal. A contract link, matching the request link, indicates that the goal has been "claimed" and is being worked on. (c) If a hypothesis to satisfy the goal is created by the schema instance on the contract link, that hypothesis is attached to the contract link and posted in STM. If further processing is possible, this information is indicated by a flag on the contract link. (d) The hypothesis is posted (asynchronously) to the instance on the request link, which receives it when it issues a command to wait on the goal. The goal node and the links between the schema instances remain as a pathway over which communication can continue as long as more processing can be done by the instance on the contract line and as long as further requests are expected from the instance on the request link. The instance on the request link is responsible for removing the goal and terminating the connections.

message is posted; that is, the flag is set to indicate that the message has been sent and the content of the message is held in a buffer associated with the goal. When the wait command is issued by the requesting instance, finding that the message is already posted, the requesting instance continues execution without waiting. If the message returned indicates that the goal is not satisfied or if the hypothesis returned is inadequate, the requesting instance can try an alternative interpretation strategy. Additional requests can be made of the responding instance through the communication link (called a contract link) between the requesting instance and the responding instance. The types of request possible are discussed under the topic of communication (Section 4.3), below.

A goal from one schema instance may cause, by the activation of another schema, the execution of an interpretation strategy that requests an additional goal. Frequently, in such a chain of requests, the second goal needs to be satisfied in order that the first be satisfied; in other words, the second goal is a subgoal of the first. Situations of this sort are characterized as "top-down" activation.

Several capabilities are a part of the design of goal-driven activation.

- A schema instance can request that one or more goals be satisfied. For example, an outdoor-scene schema instance requests goals of sky, ground-plane, and foliage. Each such goal request causes the activation of a separate instance corresponding to a goal of the requested type.
- A goal can be a request for one or more hypotheses of the specified type. For example, a window-frame schema instance could request a goal for two shutters. This would cause two schema instances to be created, both instances of the shutter schema.
- Each hypothesis requested in a goal has an associated schema instance "responsible" for supplying a hypothesis of the type requested. The instance is associated with the hypothesis request by a *contract link*, which is an arc connecting the instance and the goal. For example, when a ground-plane schema makes a request for grass, the grass schema instance that is satisfying that request is linked to the goal request by the contract link for that request. The contract link also serves as a data structure for communications (see Section 4.4).

- The schema instance satisfying a goal request *is not restricted to producing hypotheses of the requested type*. A schema instance may produce other hypotheses as a side effect of its actions, hypotheses which may (serendipitously) satisfy other goals, cause event-driven activation of other schema, or simply be left to be used later.

The second way in which a schema may be activated, event-driven activation, begins with the occurrence of a data item or hypothesis that is an important event for the schema (by virtue of the fact that it is listed on the schema's important event list). For example, a long horizontal line directly below the sky might be considered sufficient evidence for a man-made structure, a building. Since the roof of the building is most likely to be the object directly connected with the line, such an event should activate a roof schema. In this case, it would be an event on the important event list of the roof schema, and its appearance would cause the activation of that schema. If there were enough additional evidence to cause the creation and verification of the hypothesis for roof, then the roof schema instance would create such a hypothesis and put it into STM (Short Term Memory). This hypothesis might be an event on the important event list of some other schema, let us say the one for "house scene," so that when the roof hypothesis was created, the house schema would be activated. Such chaining of activation, initiated by data events, leads to the characterization of this type of activation as "bottom-up" or "data-driven." Event-driven activation can lead to an early, likely, partial interpretation, which provides a guide for the interpretation of other portions of the image through the goal-driven actions of other instances.

4.2 Schema Instance

A schema instance is created whenever a schema is activated. The instance represents the state of the application of the interpretation strategies from the schema (stored in LTM) to the image data and to the current set of hypotheses

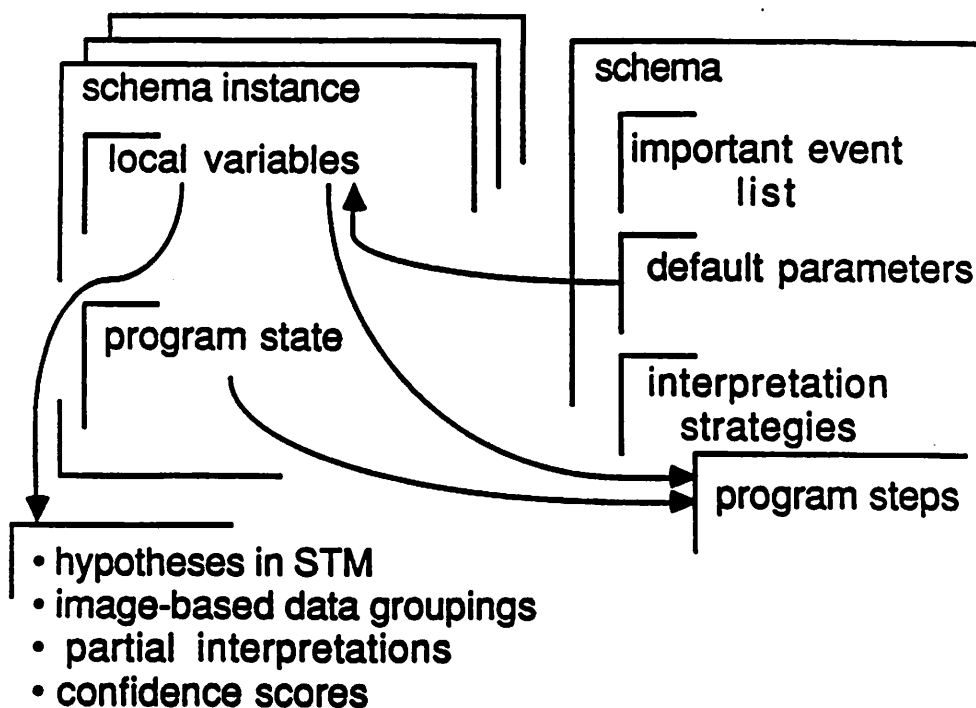


Figure 15. Schema Program and Relation to Schema Instance

The schema instance is actually the complete program state of one invocation of an interpretation strategy from the schema. Each interpretation strategy is written as a sequence of operations on a set of local variables. Thus, the schema instance records that program state by recording the values for each local variable, and a pointer into the program description of the interpretation strategy. When the schema is activated, some of the local variables receive preset values, as default values, from the schema. In addition, when a goal activates the schema, initial values for the local variables can be set from the goal. The local variables are typically used to encode intermediate information as indicated in the list here. For example, partial interpretations are built up by pointers to hypotheses in STM.

and descriptions in STM (see Figure 15). Further, it is a separate invocation of the interpretation strategies, operating independently of any other invocation.

For a given schema there can be more than one instance active at a time. The only interaction between instances of the same schema is that open to schema instances in general: they may create and wait for hypotheses or they may invoke

another schema by requesting a goal. Multiple activations would normally be associated with separate portions of the image, making parallel interpretations of separate events under the assumption that they represent different instances of the same type of object (e.g., the shutters associated with a window).

The description of the (program) state of an interpretation strategy within a schema instance includes a set of local variables. These local variables allow the interpretation strategies associated with the schema to interact closely without affecting the hypotheses in STM. The initial values for these variables come from the parameters in the schema, but those initial values can be overridden by the values of parameters with the same name in the goal on the contract link.

4.3 Communication: Messages and Shared Data

When an interpretation strategy in a schema instance requires the creation of a hypothesis, there are two options: it can directly request the creation of the hypothesis by issuing a goal or it can follow a "sit and wait" approach watching for the hypothesis to be created in STM. The first of these, the case of a direct request, was introduced in the discussion of goal-directed activation. In addition to the activation mechanism, the schema instances have the mechanism for communicating when a goal is requested: we will call this direct communication. When a goal is requested, the requesting instance can set parameters for the goal; through the formation of the contract link and the activation of a schema instance to satisfy the goal, the newly activated instance receives, as preset local variables, the parameters from the goal. In this way the requesting instance communicates details of the request to the satisfying instance. With the creation of the hypothesis, a message that the goal is satisfied is passed to the instance which originally issued the goal request. This message contains the hypothesis that satisfies the goal and which can be used to communicate information as associated parameter values.

An initial hypothesis, however, could be very tentative or there might be other conditions under which refinements might be possible. In this case, the instance that created the hypothesis indicates the possibility of refinement by setting a flag on the contract link when it responds. Then it can either wait for a request for further processing from the schema instance that requested the goal or proceed and postpone waiting. If the requesting instance requires a more refined hypothesis it signals that further processing is required. A request for further processing causes the responding instance to resume execution if it is waiting, or (if it is not waiting) to continue the next time it does wait. Subsequently, having refined the hypothesis, or perhaps having created a different hypothesis to better satisfy the goal, the satisfying instance again signals the requesting instance. This cycle of *incremental* improvement can continue as long as there are interpretation strategy steps that will make improvements. When these steps are exhausted the schema instance that is responding to the requests signals the requesting instance that there are no further refinements. The communication cycle is then terminated. With each step in the cycle, the requesting instance can set parameter values on the contract link; these values get passed on to the satisfying instance as updated parameter values. Further, the satisfying instance can communicate with the requesting instance by setting values associated with the hypothesis.

Direct communication permits interaction between schema instances. Location and other contextual information can be passed through the goal request, and once processing is started the effect of interpretation strategies can be "sampled" and reported. For example, if each schema were designed to first give a coarse interpretation, followed by a wait for a request for more processing, followed by a more detailed interpretation, then schemas composed of parts could implement a coarse interpretation based on the coarse interpretation of the parts. Thus, the

entire system could give an early coarse interpretation followed by a more detailed interpretation as time permitted.

The type of direct communication described here is a very simple type of shared memory mechanism for communication. Such mechanisms were proposed by Minsky [MIN80] as an appropriate way of communicating information between agents (read "schemas") where each agent carries out a highly specialized task. Our experiments with this type of interaction are only an example of what might be done. Much more needs to be understood about how to handle the dependencies among communicating entities before this type of mechanism can be fully exploited.

A weak argument can be made for direct communication being more efficient than other methods, such as communicating through exchanging hypotheses on a blackboard. This was not demonstrably the case with our implementation, but since it depends on the implementation of the methods this observation is inconclusive.

In contrast to this "direct" method of communicating, a means of indirect communication is provided by the other method for a schema instance to obtain a hypothesis. When an interpretation strategy in an instance requires a hypothesis, it can either wait until a hypothesis of the type desired is created in STM or poll STM for such a hypothesis and continue with the information thus gained. For example, once the ground-plane instance has constructed the ground-plane hypothesis, it waits for hypotheses of objects which should lie on the ground plane. When they occur it adds relations between them and the ground-plane hypothesis. In this way, hypotheses created by unrelated or distantly related instances can be used to communicate information. This means of indirect communication is only minimally exploited in the current system; we preferred to use the more constrained interaction of direct communication where possible. Direct communication keeps most of the control of overall processing within the interpretation strategy of the requesting

schema. The possibilities for additional types of communication links are discussed in Chapter 5. The two available types of communication proved sufficient for the interpretation strategies developed in Chapter 3.

5. Conclusion and Summary

To develop a basis for representation and control, our design of an interpretation system incorporates some of the general principles from the study of human cognition. The most important idea is that a schema is an active structure for building a description of a recognized object. Adapted in this dissertation as a framework to relate procedural and declarative knowledge, schemas represent the knowledge related to objects or scenes. Procedural knowledge is represented in interpretation strategies that:

- describe methods for object recognition;
- effectively build descriptive structures representing hypotheses that relate image data, intermediate abstractions, and other hypotheses to object identity; and
- are independently applied to separate areas of the image.

There is a schema for each object that is to be recognized by the interpretation system. By combining information describing the object color, texture, and object geometry with an active, procedural structure, the schema describes both the object and how to recognize it. Data from the image can be matched to a view in the object description providing a link between image data and three-dimensional representation. Values that can be measured from the image, those related to the object surface by region color, texture, size, shape, or location, or those related to object edge by relative length, contrast, or strength of lines in the image, are represented by parameters that fill slots in the description of the object. In addition, the description includes relations between subparts: attachment, adjacency, expected coexistence, and subpart placement. Also, there are two hierarchies based

on specialization and composition. These object descriptions contain information used by the object recognition procedures.

When an object is being recognized, there is a specific invocation of the schema, a schema instance, which describes the state of that particular recognition process. The schema instance is associated with a specific portion of the image. The activation of the schema, the creation of the schema instance, and the execution of the associated interpretation strategies creates one or more hypotheses. Each hypothesis associates regions, lines, and subordinate hypotheses with the object through a network that is constructed by the schema instance.

Schema activation creates a schema instance which indicates a tentative supposition of the existence of the object in the image. The effects of the processing associated with the confirmation of this supposition are isolated to the schema instance, but when enough evidence has been gathered the schema instance creates a hypothesis. Examples of how this process works and the discussion of the issues are covered in the next two chapters. Once this hypothesis is verified it is placed in STM (Short Term Memory). Initially, hypotheses are isolated from the total interpretation; however, the structure of the object is confirmed by the progressive association of the object with other objects in the interpretation network through the actions of other schema instances. The final hypothesis is included in the interpretation network for the whole scene.

CHAPTER III

INTERPRETATION STRATEGIES

Interpretation strategies play a fundamental role in the schema-based interpretation system. These programs represent both the knowledge needed for interpretation and the control of the recognition process for objects and scenes. They determine which actions are taken to select and group related data and hypotheses in STM to form the description of an object. Additionally, they select schemas for activation by making goal requests. In general, the interpretation strategies combine diverse object recognition methods and integrate the use of different sources of knowledge.

The types of knowledge needed for interpretation are many and varied. Recognizing any specific object requires access to knowledge about its characteristics. Much of this is "fine grained" knowledge; in other words, it covers a broad range and is fairly detailed knowledge of the appearance of the object or of the appearance of its parts and their relations. Although it may eventually be the case that such detailed knowledge can be fully represented in a general descriptive structure, given our current state of understanding, the development of such knowledge is more naturally expressed in a representation that includes the expression of *process* as well as of propositional types of knowledge. In the schema-based representation, interpretation strategies represent the processes for recognizing objects or scenes.

The determination of an appropriate strategy depends on many factors. Of course, both the interpretation task and the choices of object representation determine, in part, which types of objects are recognizable and the way in which they should be recognized. In addition, the types of data that our system is capable of extracting from the image determine what types of representations can be matched.

For example, one general goal of an interpretation task is to establish the approximate location of the object as early in the interpretation as possible. Then, for some objects, knowledge about their possible position in the scene is directly related to which parts of the object are most easily recognized. Such specific knowledge gets expressed as part of the control information in the interpretation strategy.

When designing interpretation strategies we followed the general framework of a *basic interpretation strategy* which we present in the next section. This skeletal strategy forms the basis for the five general types of interpretation strategies, each discussed in a subsequent section. Those five interpretation strategies are:

1. exemplar selection and feature matching extension,
2. exemplar selection and extension by geometric-guided construction,
3. key feature matching and extension by geometric construction,
4. key part matching and extension by part-whole relationships, and
5. context-initiated interpretation.

In developing these interpretation strategies we applied an incremental approach to testing and refining the knowledge base, similar in spirit to the knowledge engineering approach in the development of expert systems. The interpretation strategy for each schema was first developed in isolation, to a rough approximation, usually on only one image. Then the schemas were combined and the interactions of the interpretation strategies tested. Finally, a few interpretation strategies (the roof, for example) were selected for refinement and tested with the entire system on additional images. If a schema was found to be producing results consistent in quality with the other schemas in the system, it was not modified further. The process of incremental refinement of the interpretation strategies was aided by the modularity of information resulting from the use of schemas.

1. Basic Interpretation Strategy

The basic interpretation strategy is a template for the interpretation strategies discussed in the sections that follow. It is based on a cycle of hypothesis formation and verification. Methods of creating and testing hypotheses are standard in systems that interpret data; the principal difference between this system and others is the distinction between tentative, local hypotheses in the schema instance and the verified, global hypothesis in STM. The reason for this is that hypotheses are frequently formed on the basis of little evidence, but they gain support as interpretation proceeds. If early, tentative hypotheses are instantiated too soon and subsequently shown to be incorrect, withdrawing them can be difficult because all the hypotheses which depend on them are affected. During the process of interpretation, tentative hypotheses can remain attached to the schema instance as local variables and not become part of STM until verified by the interpretation strategy. The effect of having a local hypothesize-and-verify cycle both reduces the possible propagation of errors made from the untimely posting of tentative hypotheses and increases the potential for parallelism by collecting the independent actions in the early steps of the interpretation strategy. This use of local memory leads to the following basic design for interpretation strategies (the two steps marked "local" indicate those steps in which hypotheses are represented only by local variables and not in STM):

1. Generate tentative hypothesis (local)
 - a. make initial object identification
 - b. verify initial identification
2. Extend initial hypothesis (local)
 - a. extend initial object identification
 - b. verify extended hypothesis and (possibly) retry steps 1 and 2 when verification fails
3. Post hypothesis in STM (global)
4. Collect and verify related hypotheses (global)
 - a. wait for additional related hypotheses

- b. verify that they are not inconsistent with existing structure, rejecting or adjusting inconsistent hypotheses as necessary
- c. add them to description in STM or (possibly) retry with different strategy or adjusted strategy parameters on verification failure

The initial step of the basic interpretation strategy, that of generating a tentative hypothesis, performs two functions both of which serve as a "focus of attention" mechanism to limit the processing necessary for interpretation. First, the initial hypothesis serves to fix the possible locations of the object in the image. This is important because it reduces the amount of the image involved in further processing and aids in the detection of possible conflicts in interpretation. Potential conflicts are detected by overlapping areas of interest in competing schemas. Second, the initial hypothesis usually establishes a location in the image of a portion of the projection of the object. The location can be used to fix some of the image features that belong to the object. This limits the search (in feature space) for additional image features of the object and (in image space) for image regions with similar features. By making a tentative hypothesis, the interpretation strategy is given a reasonable starting place.

In the second half of the first step (1b) the interpretation process makes a test of the fitness of the hypothesis, discarding those that are found to be incorrect. In this verification step, measurements of image data are made and agreement between the hypotheses in STM and the knowledge in LTM is checked to assure that the hypothesis being formed is not inconsistent with that information. Verification can also entail scoring a hypothesis as to how well it depicts the object being described. The checks do not necessarily involve all the evidence or all the relations and descriptions in LTM; rather, they selectively apply tests that are most likely to eliminate incorrect hypotheses at minimal cost.

There are several issues which influence the complexity of the first verification phase of the strategy. Generally they are: how unreliable are the hypotheses generated likely to be, how many hypotheses are typically generated, what is the cost of generating a hypothesis, should the verification be applied to all the hypotheses generated, and how costly is the verification test? These issues can be summarized in the first of two trade-offs. The designer of an interpretation strategy must choose between placing complexity in the generation routine (step 1a) or in the verification routine (step 1b). In the first case, it is possible to limit the number of unreliable hypotheses by increasing the sophistication of the hypothesis-forming process. This results in the generation of fewer hypotheses, thus requiring less testing in the verification phase. In the second case, when the designer chooses a simpler generation process, far more unreliable hypotheses may be generated, requiring a more complex verification process. By a judicious choice of verification procedures a balance is possible in which the requirements of the hypothesis-generation phase are simplified. For example, as presented later in the chapter, the roof interpretation strategy generated a tentative hypothesis based primarily on a measurement of simple image features, such as color, texture and size. The simplicity is desirable because there are times when these measures must be made over all the regions of the image. However, because relatively few regions are put forth as possible hypotheses, verification can utilize more complex measurements.

Another aspect of the trade-off – that is, whether to place complexity in hypothesis generation or verification – is that, in general, the later in the interpretation process that the verification phase occurs, the more knowledge about the object it has available, and the more costly the respective test can be. As the interpretation proceeds and partial interpretations are built up, increasingly more of the

relations that can be used to verify the existence of the object will have their corresponding hypotheses posted. The verification procedure that is placed later in the interpretation process can rely more on the possible existence of these hypotheses.

When a hypothesis is rejected by a verification test, there are several alternatives for generating alternate candidate hypotheses. The interpretation strategy can attempt to repeat the strategy step at which the hypothesis was rejected with relaxed constraints (i.e., widening the search and possibly accepting less reliable hypotheses), an alternate interpretation strategy can be tried, or the interpretation strategy can report a failure and quit or wait to be signaled to retry. Finally, any combination of these responses can be used.

In the second step of the basic interpretation strategy, routines are called to extend the hypothesis. Often the regions and lines associated with the initial hypothesis will provide clues as to which image features are useful in searching for data to confirm the object. As additional information is gathered, structure is added to the initial hypothesis, creating a more complete description for the object as it appears in the image. The hypothesis formed in this second step is also verified. Thus, the initial hypothesis is used as a guide and the description of the object is extended.

Once the interpretation strategy has a more certain and complete internal hypothesis, that hypothesis is posted in Short Term Memory. In the general plan described here, this posting is done once, as the third step of the basic interpretation strategy. In so doing, it suggests a solution for the second trade-off that the designer must consider, that of choosing between early and late posting. While early posting makes object hypotheses available for other processes sooner, it is more likely to lead to a propagation of interpretations dependent on an erroneous

hypothesis. On the other hand, late posting exhibits greater local consistency because of repeated verifications; therefore, the hypothesis posted is more likely to be correct. In the specific implementations that follow, posting is often done both at this point in the strategy and at the earlier points during the hypothesis extension. The factors going into this decision are really domain dependent and are the results of our design choices. The general principle which we used was our expectation of the reliability of the results of the first stage. For example, the initial region selected for sky is often the largest part of the sky in the image; because of this expectation, the interpretation strategy for sky posts the tentative hypothesis in STM at the end of the first step, in addition to posting the extended and verified hypothesis at the end of the second step. Additional verification measures can be made to assure the certainty of the hypothesis that is to be posted.

After posting the hypothesis in STM, the schema instance can remain active. In this case, it will continue to fill in detail, collecting and adding information to the hypotheses which it posted. For example, after posting the ground-plane hypothesis, the ground-plane schema instance remains active and waits for (optional) objects that might occur on the ground plane. When hypotheses for those objects are posted in STM, and if they are not inconsistent with the ground plane hypothesis (i.e., the one associated with the schema instance), they are added to the ground-plane hypothesis in STM as parts of the ground plane.

The design of the verification phase in step 4b involves some complex issues. Generally, inconsistencies at this stage are more complex and involve a greater number of past decisions, some of which were made by other interpretation strategies, perhaps much earlier in the course of the interpretation. Since the interpretation strategy is limited to controlling the local interpretation, and (indirectly) other strategies with which it is communicating, correcting errors and inconsistencies at

this stage can be difficult. General methods for detecting inconsistencies and deciding what to do when they occur are areas of active research (for examples in computer vision see [NAG80] for a method for using backtracking, [TSO84] for a method using compatibility measures, and [WES86] for a method using evidential reasoning). However, none of the general methods are easily applied to systems where a large portion of the knowledge is expressed procedurally. Therefore, our system relies on specialized verification methods such as the ground-plane strategy steps described above.

In the sections that follow we will examine implementations of and variations upon this basic interpretation strategy. The schemas in our system can be grouped by the particular variation of interpretation strategy used, and they are listed below in the order in which they are discussed in subsequent sections:

- Exemplar selection and feature matching extension
 - Grass, Sky, and Foliage
- Extension by geometric-guided construction
 - Shutters, Road, and Roof (first strategy)
- Key feature matching and geometric construction
 - Roof (second strategy)
- Key parts and part-whole relations
 - Ground-Plane, House Scene, Outdoor Scene, House, and Walls
- Context-initiated interpretation
 - Wire and Telephone-Pole

This dissertation is about a method and framework for designing an interpretation system. Much of the work for this first interpretation system is experimental in nature. While developing the framework to allow the concurrent interpretation by several schemas, we also had to develop interpretation strategies to use within that framework. Each of the strategies discussed in the sections that follow is a snapshot of the process of developing an interpretation program. Therefore, while some of these strategies are relatively primitive and image dependent, others illustrate the direction that can be taken to create interpretation processes that are independent

of the data. The effort to make more robust and general schemas requires the incremental refinement of the knowledge by testing in the intended domain.

2. Exemplar Selection and Feature Matching Extension

The "exemplar selection and extension" strategy consists of three steps. In terms of the basic interpretation strategy described earlier, it makes a tentative hypothesis for the object of the schema, extends that hypothesis, and posts it in STM. In the first step, an "exemplar" region is selected from the image and labeled as belonging to the image of the object. From this labeled region a simple hypothesis is created, stating, in effect, that the object occurs in the image at the location of the labeled region; this is the tentative, internal hypothesis. In the second step, selected features of the exemplar region are compared to those of other regions in the image to select a more complete set of regions corresponding to the projection of the object. These regions are added to the hypothesis. The third step is the posting of the hypothesis to STM. These strategies have no fourth step, because in the set of schemas developed for these experiments there are no hypotheses which contribute further to the interpretation of these objects. The interpretation strategies for the grass, sky and foliage schemas are of this type.

Verification for this type of strategy in the local and global steps can come from many sources. Since the strategy steps for hypothesis formation and extension are based on image-related data, and since they deal with objects having little or no description of geometric structure, the best source of verification information is in the relations in LTM. The local verification (step 1b) could be based on size and shape consistency; for example, the hypothesis for grass could be checked to see that all the grass regions were in roughly the same part of the image. Further, a possible method for global verification of the hypothesis in STM (step 2b) would use inter-object relations. For example, a hypothesis for sky could be verified by a procedure to check if the sky were above the grass based upon an assumption of an upright

camera (or referring to a camera model). While possible, verification steps such as these were not used in this implementation because the interpretation strategies which we developed did not exhibit any dramatic failures in the interpretations attempted; however, in a broader test of this system, more powerful verification strategies undoubtedly would be necessary.

2.1 Grass - Feature-Based Exemplar Selection

The two most straightforward exemplar selection and extension strategies are those used by the grass and sky schemas. The exemplar is selected by a rule-based labeling function [HAN85]. As described below, this function computes a weighted sum of scaled differences between the feature range descriptions in the schema and the feature values of regions in the image. Those regions which are assigned a (non-veto) score are considered as candidate exemplars and the region having the highest score is selected as the first exemplar for consideration. In some strategies (e.g., grass) only the first exemplar is used, while in others (e.g., the first roof strategy) the entire candidate list comes into play.

A system has been designed for the development and evaluation of exemplar selection measures and is discussed in [HAN85]. Nevertheless, we want to outline the essence of this method for exemplar selection. For the grass schema, the ideal exemplar region should be one that is typical of grass regions in the particular image; in other words, the grass exemplar should be selected from among those that would be labeled "grass" if all regions were correctly labeled. In order to select such regions we attempt to discover which image features tend to be more invariant (over a set of images) for grass regions; these features and their values provide a description of the typical grass region. For example, grass is green and textured; therefore, we should be able to find "greenness" feature values and "texture" feature values that are typical of grass. Once such typical feature values are determined, they become part of the description of "grass" in the grass schema.

We express the typical features for an object as feature value ranges captured by a scoring function ([HAN85]). The motivation for representing the feature values as ranges can be seen in the histogram shown in Figure 16. When looking at the data from a large number of grass regions (from several images), we can see that the area in the histogram corresponding to the object covers a wide range of the feature value. It is not easy to pick one single typical value; in fact, such an approach is unreasonable. It is better to express our uncertainty as to the typical value

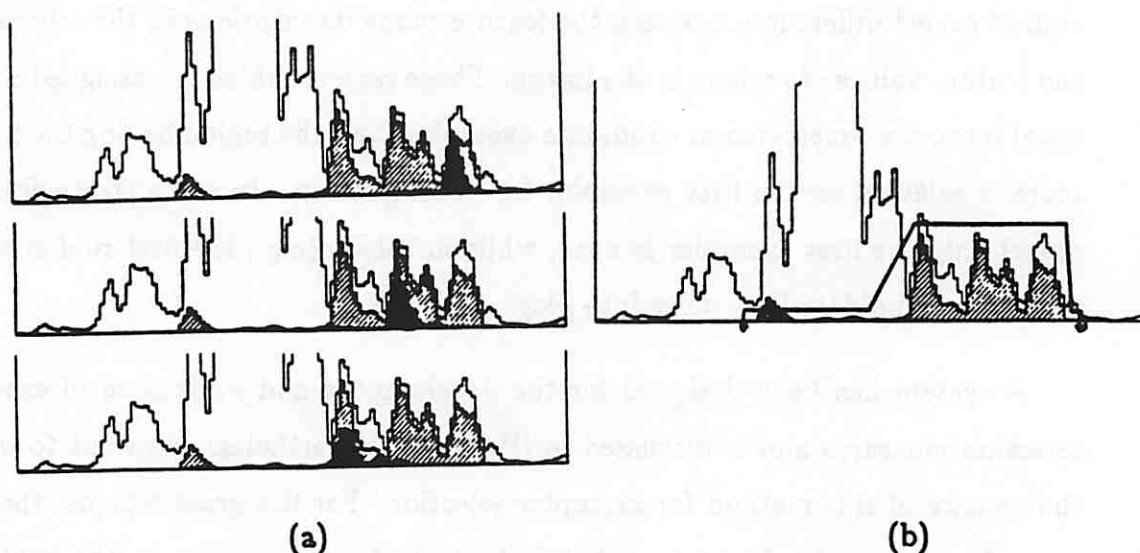


Figure 16. Histogram-Based Feature Scoring Function

(a) The overlapping histograms of an "excess green" feature (2G-R-B) were computed across eight sample images. The unshaded histogram represents the global distribution of this feature. The cross-hatched histogram is the distribution of the feature in hand-labeled regions known to be grass across the same set of images. The dark histogram shows the feature distribution for one image. Note the shift in the distribution of the feature from image to image. (b) An exemplar function is superimposed on the histograms. The independent axis encodes the feature value, increasing to the right. The feature value is transformed into a score (the dependent axis) by this function. See the text for further details.

by stating that the value can fall within a range. When a feature value falls within the typical range, the feature "votes" for the object by contributing towards a score. Further, because a small change in the feature value should not result in a large change in its contribution to the score for the object, the contribution of the feature to the score tapers off linearly at the ends of the range. Thus, the relations between the feature-value ranges and the object (for example, the relation between excess-green and grass) are represented with a scoring function described by six control points, as illustrated in Figure 16b. The score for a particular feature is maximal and undifferentiating in the range of feature values that are typical of grass, and it is minimal in areas where the feature value does not distinguish between the target object and other objects. In addition to the score value, potentially each feature can contribute a "veto" vote. If any particular feature is outside a reasonable range, then the score for that feature carries a "veto" and the final score for the region is a "veto." More precisely, the contribution of the feature value to the score of a region is determined by the following function for a feature f , based on the six control points, denoted as θ_i , $i = 1 \dots 6$:

$$score_f = \begin{cases} \text{veto,} & \text{if } f \leq \theta_1 \\ 0, & \text{if } \theta_1 < f \leq \theta_2 \\ 1 - \frac{\theta_3 - f}{\theta_3 - \theta_2} & \text{if } \theta_2 < f \leq \theta_3 \\ 1, & \text{if } \theta_3 < f \leq \theta_4 \\ \frac{\theta_5 - f}{\theta_5 - \theta_4} & \text{if } \theta_4 < f \leq \theta_5 \\ 0, & \text{if } \theta_5 < f \leq \theta_6 \\ \text{veto,} & \text{if } \theta_6 < f \end{cases}$$

The scoring function using six control points is the full form of the function. It is called a "simple" rule to distinguish it from those rules, discussed shortly, in which the contributions of several features are simultaneously considered. The threshold parameters can be omitted to make "open ended" functions. The omission of either

veto parameter makes a function that tails off in a minimum score rather than supplying a veto vote. The omission of the first three (or last three) parameters makes a function that tails off at a maximum value on one side and at a minimum value (or veto) on the other. These forms are summarized in Figure 17.

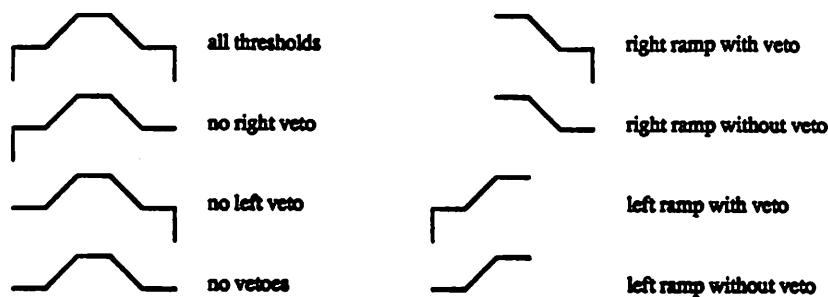


Figure 17. Forms of Scoring Functions

These various forms of the scoring function can be achieved by selectively including or excluding the control points (θ_i , $i = 1 \dots 6$) from the scoring function given in the text.

To use the scoring functions it is necessary to pick features which discriminate the target object from other objects. We used a verbal description of the features of the object as a first approximation of the appropriate feature values. For example the green of grass was described as a "medium to high value of excess-green." The feature measures used in the verbal descriptions (i.e., excess-green) were drawn from a set of feature measures developed by Williams and Nagin [WIL81]. The selected features were tested in combinations to see if they would select grass regions in the images of our test set. When we found features that were obviously detracting from contributions to the score, they were discarded or altered. The features used in the scoring rules discussed in this chapter are given in Table 1.

Table 1.
Features Used in the Scoring Rules.

Red, Green, and Blue.....	the raw color value from the image (R, G, and B).
Excess-Green.....	a measure of greenness, also called green-magenta opponent color (2G-R-B).
Excess-Blue.....	a measure of blueness, also called blue-yellow opponent color (2B-R-G).
Excess-Red.....	a measure of redness, also called red-cyan opponent color (2R-G-B).
Intensity.....	grey value $I=(R+G+B)/3$.
Normalized Red	$NR=R/(R+G+B)$.
Normalized Green.....	$NG=G/(R+G+B)$.
Normalized Blue.....	$NB=B/(R+G+B)$.
Location of Bottom.....	the vertical position of the bottom extent of the region.
Location of Top.....	the vertical position of the top extent of the region.
Location of Centroid.....	the vertical position of the centroid of the region.
Width.....	the distance between the left and right extents of the region. (Note that this is given as a size feature and not a shape feature.)
Size.....	number of pixels covered by the regions (as a ratio of the size of the image).
Compactness.....	the ratio of size to area of enclosing rectangle. (Note that this is really a "rectangular" compactness measure. It was, however, sufficient.)
Height-to-width-ratio.....	the ratio of the sides of the enclosing rectangle; sometimes given as width to height ratio.
Saturation.....	a measure which approximates the strength of the color, $S=1-\min(NR,NG,NB)$.
Short Line Density.....	a texture measure, the ratio of the number of short lines to region area; a short line is a line for the straight edge finding algorithm (described in Chapter 2) with a length shorter than 5 pixels.
Extremum Count.....	a texture measure; the ratio of number of extremum pixels (i.e., those central pixels with the maximum or minimum value in a local neighborhood) to the size of the region; intensity values were used.
Standard Deviation of Color Values	the statistical variance of color and intensity values is an additional texture measure.

Note that when any feature value is used, it is normalized by the object-specific scoring function for that feature.

Discovering which features were non-discriminatory proved to be difficult. Some features that seemed to make sense on the basis of the verbal approximations proved, when examined later, not to contribute to the final results of discriminating grass. The examination of single features and their corresponding functions was problematic. While selecting features that clearly discriminated grass from other objects was easy, determining which features did not discriminate was much more difficult. The examination of the results of single functions did not reveal the features that were working in pairs to discriminate the object. Further, if one looked at a histogram or the image of the feature values, some features that appeared not to be discriminating the object were important because they would be eliminating (by veto) regions that were otherwise difficult to discard.

Despite its failings, this method for selecting measures proved sufficient to develop the scoring functions that were used in this dissertation. Also, starting with symbolic descriptions illustrates the utility of a language-like description in assigning features for objects. Because of the naturalness of symbolic expressions, it is possible to develop the functions very rapidly. The prototypes in this dissertation are used as rough approximations to scoring functions which could ultimately have been developed by a more time-consuming, closer examination of the data; thus, our time and energy were allocated to the development of other parts of the system.

Concurrent with the research described in this dissertation, there is an ongoing effort to develop tools and methods for the discovery of the correct features to use in rules of this sort. With the system now developed (see [HAN85]), using a modified expert-systems approach, an experimenter can interactively examine the distributions of several features, sketch rules, and display the resulting labelings. Although still very labor-intensive, the methods now available are much better than the ones used as a basis for the rules described herein. For example, Figure 18 shows a display generated by the new system illustrating that some of the features chosen

Figure 18. Feature Histograms with Scoring Functions

A display from the newly developed system for investigating the utility of feature values. These plots of feature histograms help in the process of developing rules for object labeling [HAN85]. From left to right, top to bottom the histograms show the distribution of: number of labeled pixels across the bottom of the picture, compactness, "excess green" ($= 2G - R - B$), extreme-value count, horizontal-edge strength per unit area, mean intensity, area, position (row in picture) of region bottom extent, position (row) of region top extent, vertical-edge strength per unit area, and the width of region in pixels. Each histogram shows the number of regions having particular values of the feature. All the histograms are smoothed.

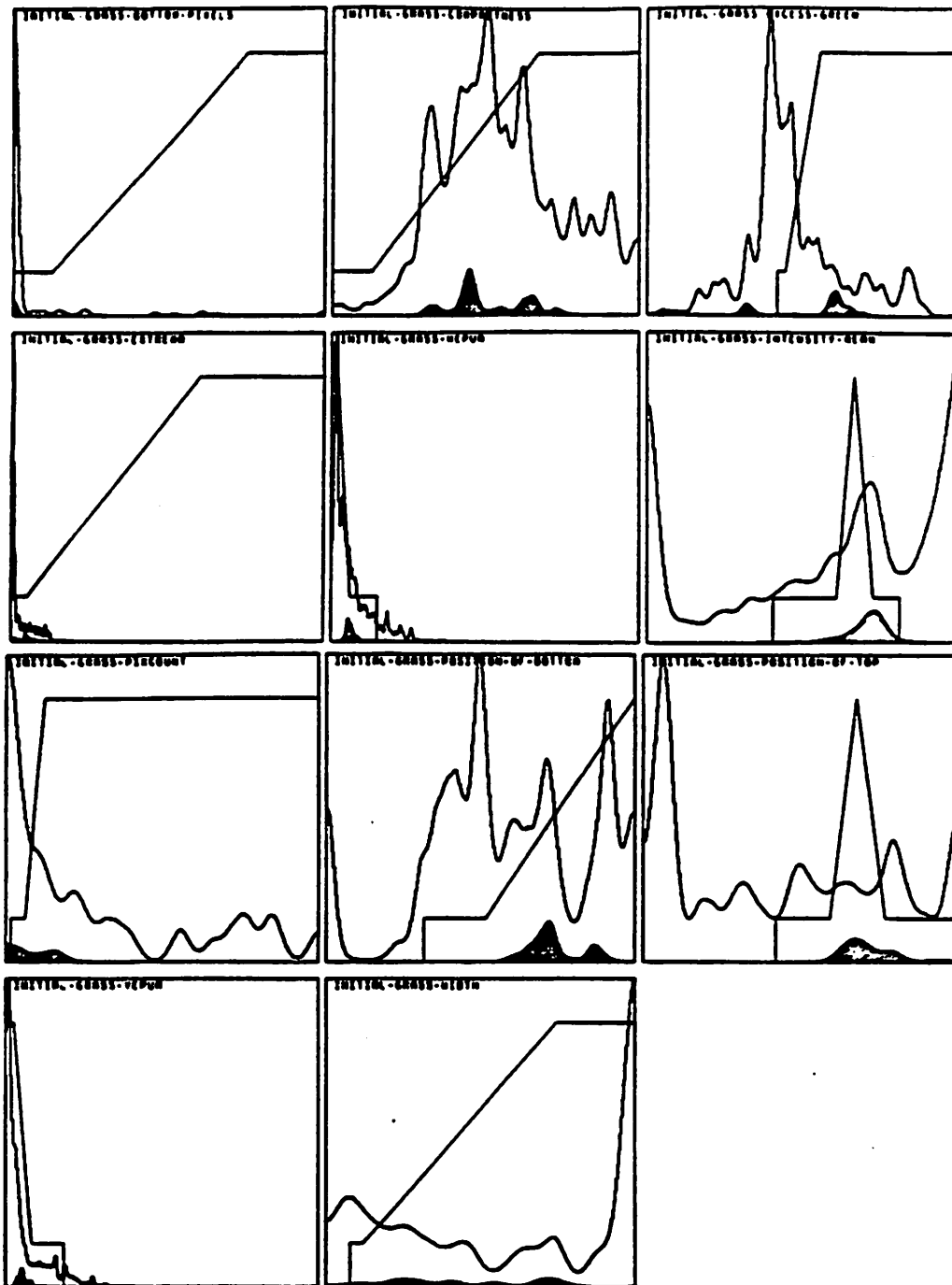


Figure 18.

by starting with the symbolic approximations are not effective in selecting grass regions.

Complex rules combine the scores for several features; the scores from individual rules are combined as weighted averages in five general categories: size, shape, color, texture, and location. The scores for each category are also combined by a weighted sum. The combined score rank-orders the non-vetoed regions, with the region having the highest score presumably being the most "grass-like."

Figure 19 shows a summary representation of the combination function for grass. The function is a weighted average of scores from each of the five categories of features with the relative weights for each category indicated by the number in the box above the category label. In each category one or more object-specific feature scores contribute to the category score. If there are contributions from more than one feature, the scores are combined by averaging them. Each feature in a category contributes equally unless otherwise indicated. (See the figure for the foliage rule, Figure 24, for an example of weighted averaging on the contributing features.) The approximate ranges of the feature values which correspond to a high score is summarized on the figure. Thus, for example, the feature "medium intensity" in the contribution to the color score is reflected in a scoring function that scores high when the region's mean intensity measure ($= (R + G + B)/3$) is in its medium range.

All feature measures are made relative to a scale that is part of the knowledge base. For example, the range of intensity values that is reasonable for the image of an object is expressed as a ratio of the difference between the maximum and minimum intensity for the entire image. Similar ratios are used for distance and size measures.

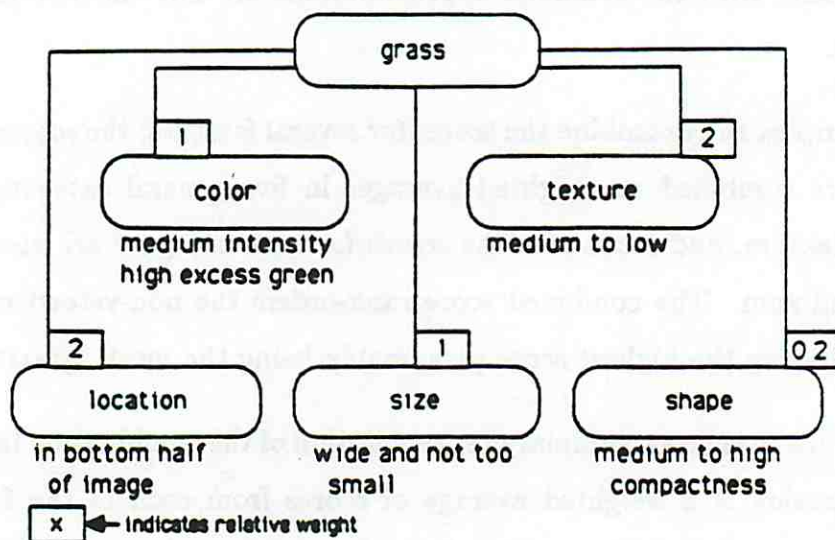


Figure 19. Combination Function for Grass Exemplar

This is a summarizing representation of the scoring functions combined to score each region as a "grass exemplar." Each branch represents a combination score for one of the categories: color, texture, location, size, and shape. The lines of text under the node for a category indicate the individual features that were combined for that category. Unless otherwise indicated the primitive feature scores for a category were averaged with a relative weight of one. (See the scoring function for foliage for an example of using other weightings.) Thus, for example, the color feature score for grass is derived from the average of scores for intensity and "excess-green." See the text for further discussion.

The scores from the features are combined to make a score for the feature category, and the category scores are combined to give a score for the object label. For the grass exemplar rule, the region with the highest score is selected as the exemplar region and it supplies an initial placement of the location of grass in the image (see Figure 20a).

2.2 Grass - Exemplar-Based Hypothesis Extension

In the exemplar selection and extension strategy, a weighted sum of scores from image-based measurements provides a rank-ordering of regions in the image. The region with the highest score is selected and serves as an exemplar region for the

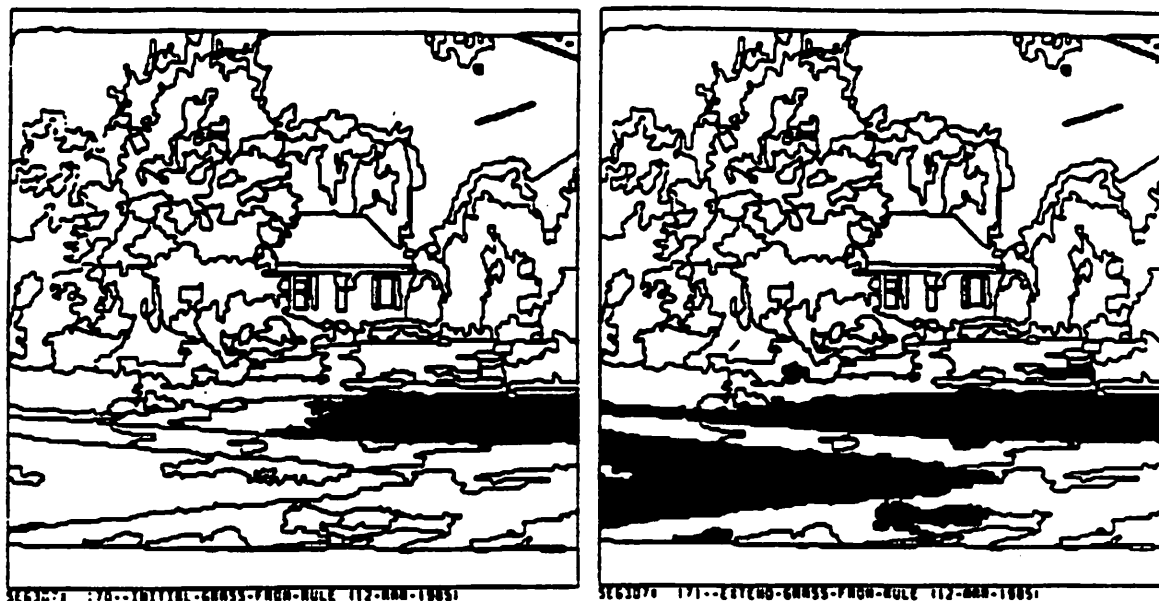


Figure 20. Exemplar Selection and Extension for Grass

- (a) An exemplar region is selected from the feature scoring rule for grass.
- (b) The features of the exemplar region are used to select additional grass regions.

search for additional regions in the image. Additional regions are then selected by matching with the features of the selected region, and the set of selected regions is combined into the image area associated with the hypothesis for the object. Exemplar selection and extension interpretation strategies illustrate the idea of starting from a reasonably certain "seed" hypothesis and extending it by gathering additional data based on the features of the initial hypothesis.

The selection of a grass exemplar makes available image-specific knowledge about the "typical grass region." The features of the exemplar region, including its location in the image, can then be used to select additional regions as grass. The features of the exemplar region are compared to the features of other regions in the image using a weighted sum of the absolute value of difference between the features,

providing a difference score which is used to select other grass regions (Figure 20b). These are grouped to form the image area for the object (grass) and are associated with the grass hypothesis in STM.

We made two design decisions that need to be examined here. First, we had the problem of determining which features to use in the computation of the difference scores. It would be possible to use the same features which served in computing the grass exemplar score; after all, these were selected as "grass-like" features. However, the problem of selecting the extension regions is different from the problem of selecting the exemplar region; that is, in selecting the exemplar region it is better to disregard (veto) grass regions that were non-typical rather than run the risk of admitting non-grass regions with grass-like features, while extending the exemplar requires a measure that is more likely to catch similar regions. We found a narrower selection of features a better choice; in practice, we used a simple color and texture difference measure.

The second design choice revolves around the question of how to use the difference score to extend the hypothesis. Because it effectively rank-orders candidates for hypothesis extension, the difference score could be part of an iterative process. For example, one implementation of step 2 (the extension step in the general interpretation strategy template) would be to take the first extension candidate and attempt to verify the resulting hypothesis. If that were verified, the next candidate region could be added and the process could continue searching down the candidate list until some reasonable stopping requirement were satisfied, thus building a reasonable extension. Unfortunately, this method puts a heavy burden on the design of the verification step. Again, we found that a much simpler method was sufficient for our investigation: all the candidate regions with difference values under an empirically determined threshold were selected and labeled as grass (see Figure 20b).

2.3 Sky and Foliage - Additional Exemplar Extension Strategies

Two additional interpretation strategies, those for sky and foliage, illustrate further application of the exemplar selection and extension strategies. Both of these interpretation strategies follow the same pattern described for the grass strategy; they differ only in the features used for the selection of the exemplar and in the extension stage.

The sky strategy illustrates the interaction between the segmentation routines and the needs of an interpretation strategy. In the images we worked with, because the sky is uniformly bright, saturated, and without much texture, it was segmented as one or two large regions. Thus, the feature-based scoring rule can rely on the segmentation routine to separate the sky region, and the size of the region can be used as an additional feature of the sky. This interaction between segmentation and feature measures makes the sky regions easy to locate in the image. Further, because of the ease of segmentation the sky is frequently segmented as one region. In this case, the exemplar is that region. In other cases it is also easy to pick out additional sky regions based on similarity of color and brightness. In more complex situations (where sky might have significant intensity and color variation due to clouds or sunset, for example), there would need to be a correspondingly more complex interpretation strategy. Figure 21 shows an example of the results of an interpretation of sky. Figure 22 gives the combination exemplar selection function for sky.

Figure 23 illustrates the results of the interpretation strategy for foliage; the combination function for foliage appears in Figure 24. Generally, the results shown for foliage are less satisfactory than the results for sky and grass. Design problems were encountered because it is more difficult to identify features that clearly separate foliage from other objects. Many of the features that can be used to select foliage are affected by the broad range of color and texture typical of foliage regions. It

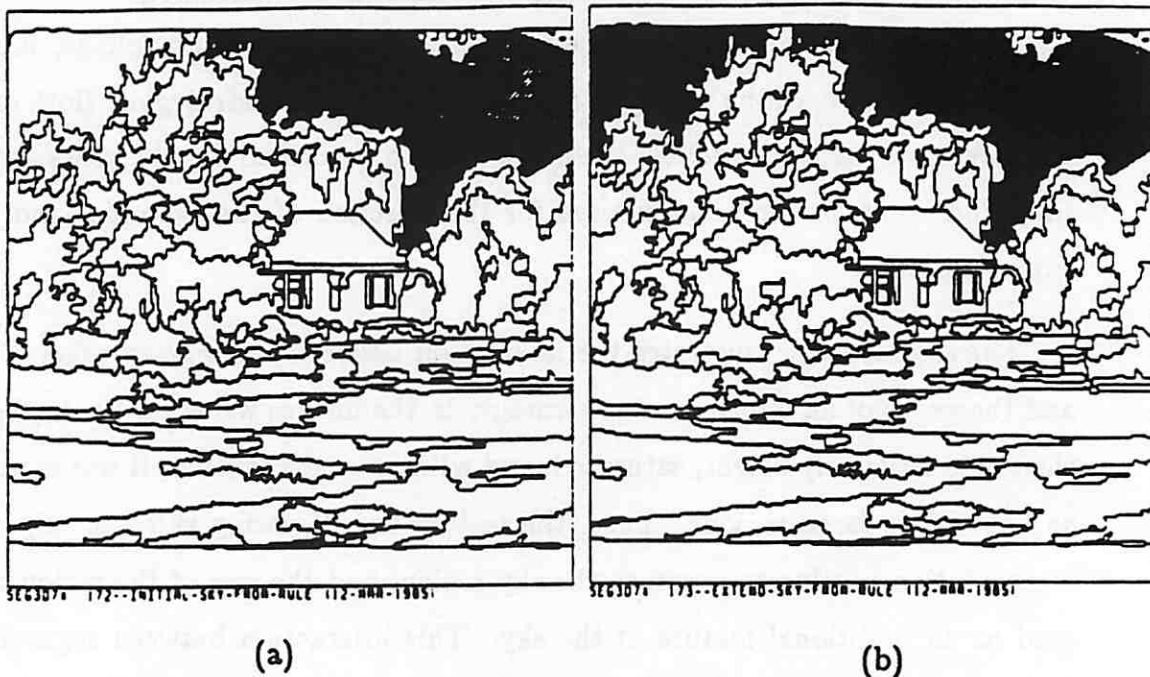


Figure 21. Sky Interpretation Using Exemplar Selection and Extension
 (a) An exemplar region for sky is selected and (b) extended by feature matching.

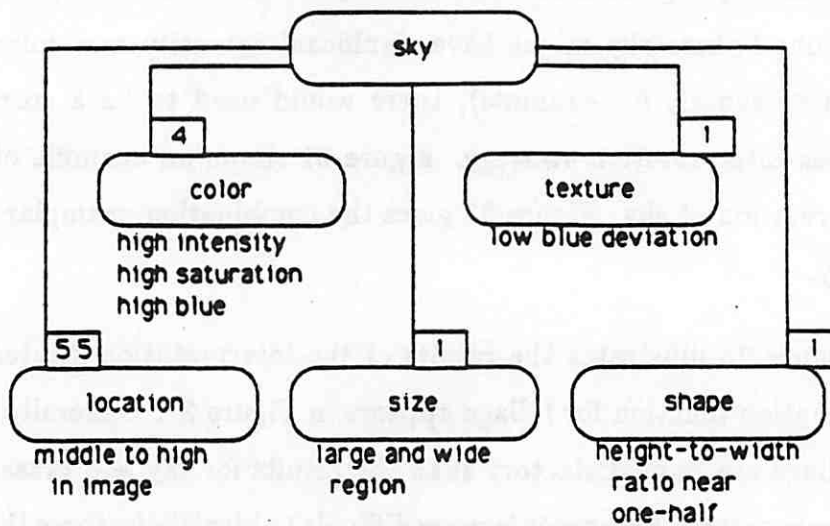


Figure 22. Combination Function for Sky
 The features used in the exemplar selection rule for sky are shown.

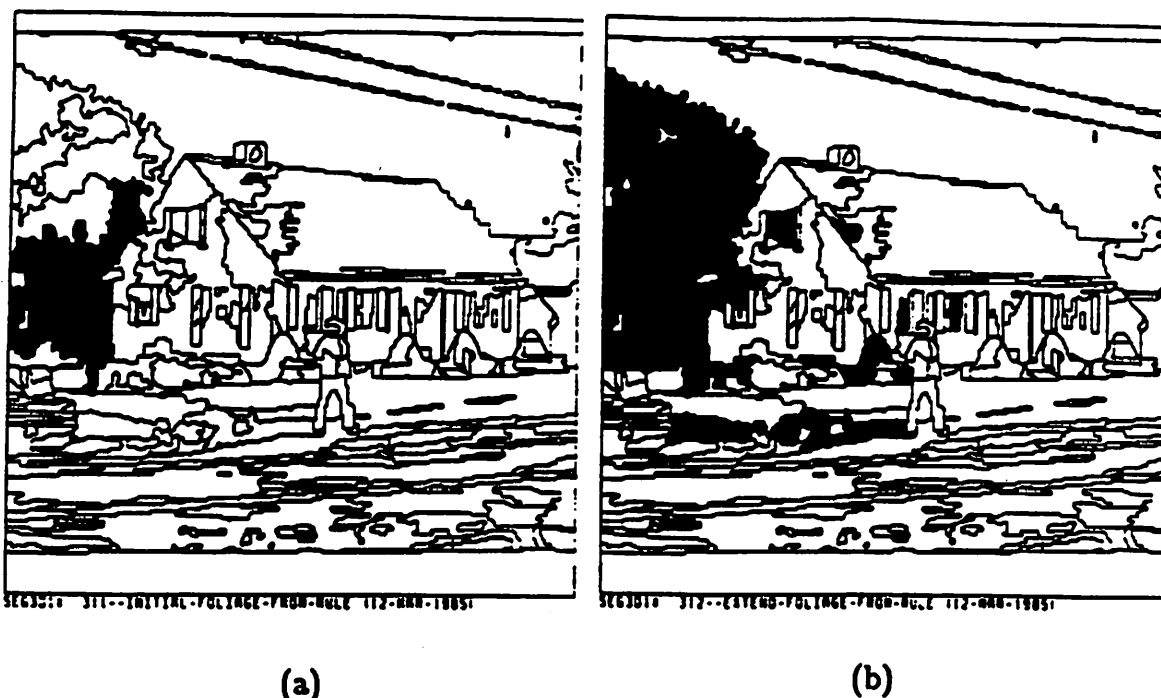


Figure 23. Foliage Interpretation Using Exemplar Selection and Extension
 (a) An exemplar for foliage and (b) its extension using feature matching.
 Note errors that result from texture-like features of nearby objects.

would seem that the effect of this variation could be countered by widening the ranges on the rules, but this causes regions corresponding to other objects to be selected. On the other hand, narrowing the range causes regions of foliage to be discarded. The present rule is a compromise between these extremes.

The reader may well note that some of the features used require stronger assumptions about the type of region being sought than others. For example, for the shape and location features, we started from the assumption that the best foliage exemplar would be one that was central, round, and reasonably large. Thus the location features shown in Figure 24 are specific to a range of viewpoints. For example, the score on the feature of the bottom extent of the region is derived from the assumption that the exemplar will be near the horizon. Because we assume

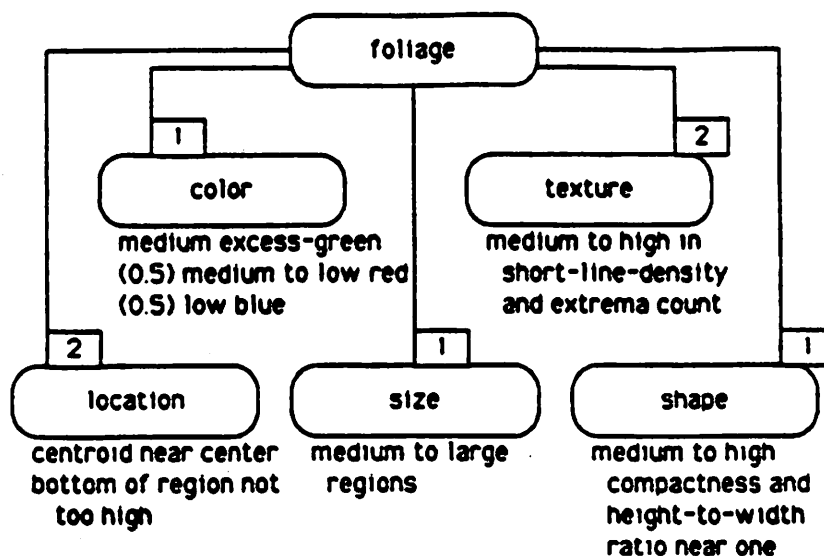


Figure 24. Combination Function for Foliage

Features used for the foliage exemplar selection rule. Under the category "color" the numbers in parentheses indicate the relative weight of those features. The color score is obtained by averaging the scores for excess-blue, red, and blue with a relative weight of 1, 0.5, and 0.5 respectively.

that the camera is upright and level this means that the regions should appear "in the middle" of the image. While these assumptions are viewpoint dependent, they hold over a wide range of viewpoints.

In this example, many of the regions selected by exemplar extension are not foliage regions. The regions in the house are either painted black with textured surfaces (the shutters) or are in the shadow of the tree. Thus, those regions are dark and have a texture measure similar to that of the foliage exemplar. The grass regions were selected because they contain tall uncut grass in shadow which has many of the same color and texture characteristics as foliage.

The mislabeling of regions often leads to multiple labels for those regions. This is one type of interpretation conflict. The resolution of such conflict is handled, in the schema network, by the outdoor-scene schema. In this case, the final interpretation

of the regions in conflict is determined by an estimate of the distance of the objects from the camera. Other heuristics are possible; spatial and shape relations could be used. Ideally, the resolution of this type of conflict should account for occlusion and the apparent partial transparency of clumps of leaves. We concentrated on only one aspect of this: occlusion. In fact, the general problems of occlusion are avoided by using a simple rule. First, each foliage region was first classified by its relation to the horizon. Then, foliage regions above the horizon were assumed to be tree foliage and were assigned to the background; that is, they were arbitrarily given a large distance. The effect of this was to give precedence to any alternate interpretation that occupied the same portion of the image (for example, where those foliage regions above the horizon overlap the parts of the house in Figure 23a, the house interpretation takes precedence). On the other hand, for foliage regions which extended below the horizon (shrubs, for example), a heuristic embodying more knowledge was used. These foliage regions were assumed to be standing on the ground plane and were given an estimated distance based on their bottom extents as described in the discussion of the ground plane interpretation strategy below. The position of the horizon was determined by a fixed camera model (assuming an upright, level camera) and set at the midline. Since these regions have associated distances they may, in fact, take precedence over other interpretations.

3. Extension by Geometric-Guided Construction

Geometric models can be used to extend an initial hypothesis derived from an exemplar region to a more complete hypothesis for the object. The interpretation strategies discussed in this section highlight the application of geometric relations in combination with other relations. We will examine the strategies of three schemas: shutters, road, and roof. (This specific roof strategy is only one of two employed; the second roof strategy will be discussed in the next section.) Using the initial hypothesis as a base, each of these interpretation strategies applies information

about the geometric structure of the object to extend the hypothesis by alternating between regions and straight lines, and by applying heuristics based on geometric expectations to accumulate evidence for the object. By matching object features to geometric features in the model we are able to construct a description of the object in the scene.

3.1 Shutters - Coalescing Fragments

We will begin with the "shutters" interpretation strategy. The interpretation strategy for the shutters schema illustrates how an initial set of regions can be extended using spatial and color constraints to indicate the location of shutters in the image. The extension is guided by specific information about the expected appearance of shutters in the image. The first step for this interpretation strategy is to select a set of regions having a high likelihood of being shutters or fragments of shutters. This is done with an application of a rule-based labeling function similar in form to the ones discussed in the last section. The application differs in that the function selects a set of regions rather than a single exemplar. The vetoed regions are discarded and the set of non-vetoed regions constitutes the initial interpretation.

The scoring function for shutters (Figure 25) is another example of the use of viewpoint-dependent assumptions. The scores reflect the initial estimate of how shutter-like each region is. The measures express the idea that "shutters are vertical rectangular regions that are not too large;" the features used are rectangularity, region size, and height-to-width ratio. These features are part of the description of shutters; that is, they spell out how a shutter is expected to appear in the image. For this strategy, we have assumed that the camera is upright, level, and a reasonable distance from the wall on which the shutters appear. Thus, the feature "rectangularity" is a measure of an "upright rectangle" in the image; we measure the similarity of the region shape to that of a rectangle.

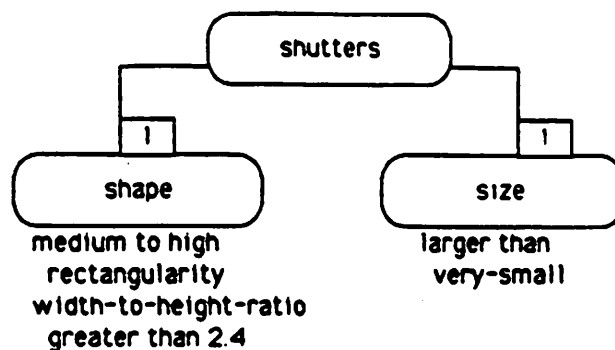


Figure 25. Combination Rule for Shutters

This combination scoring function is used to select starting region and to measure region groupings for "shutter-likeness."

The measure of rectangularity, together with the assumption of distance from the object, determines the minimum size restriction. The smaller the region (regardless of shape), the more rectangular it is; thus any region that is small enough will have a good fit to a rectangle in the image. At the extreme, one- and two-pixel regions are always perfect rectangles. Thus, small regions must be excluded from consideration.

In the extension step (step 2) of this strategy the set of regions selected by the scoring function constitutes a collection of "seed regions" (Figure 26a). The figure shows two types of problems with these sets of seed regions: some regions (shadows, etc.) are incorrectly labeled shutters and some shutters are incomplete and fragmented. To complete the "unfinished" shutters, the initial regions are partitioned into clusters of adjacent regions (Figure 26b). Each cluster of regions serves as a starting point for a search for missing regions. They are combined with adjacent regions (Figure 26c) and that set of adjacent regions, including the original seed regions, supplies a list of subsets, each of which is tested to determine if it fits the description of a shutter. The method of testing is this: each subset of the set of the adjacent regions that includes any of the original regions is temporarily treated as a

Figure 26. Interpretation Strategy for Shutters

(a) The seed regions that were selected by the shutter scoring function. (b) Shows one cluster from among clusterings of initial regions. (c) Additional regions are added to the cluster from (b) to form the image area considered in identifying the final shutter. (d) Shows the final shutter formed by searching the candidates shown in (c). (e) Shows a schematic representation of the constraints used to select shutter pairs. (f) Shows the shutter pair which includes the shutter shown in (d).

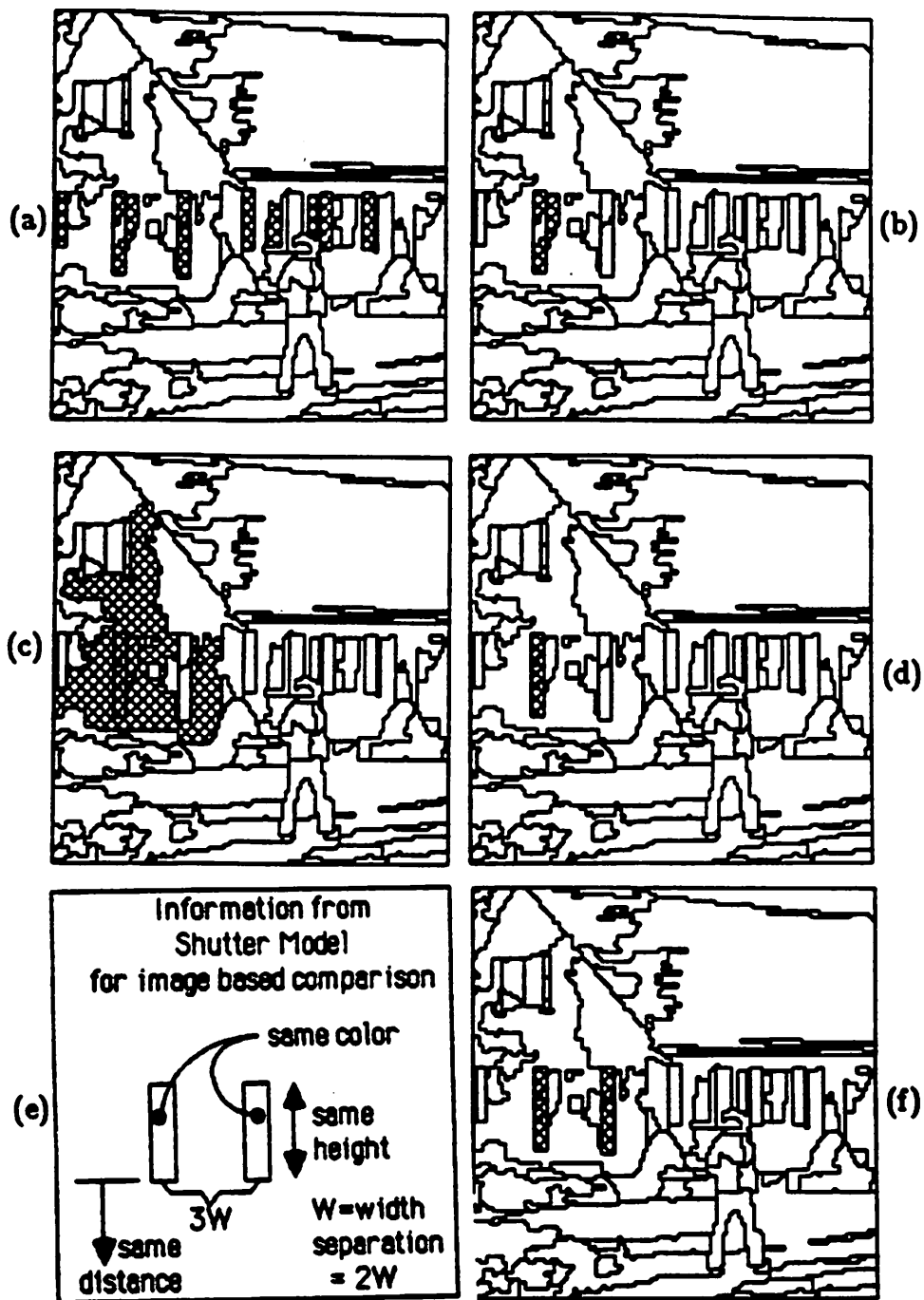


Figure 26.

single region and passed to the function that makes the measurement for the shutter rule (e.g., the rule used to select the initial shutter). Of those, the subset with the highest score is assumed to define the best fit to a shutter. The regions of that subset are grouped together and labeled as an initial shutter (Figure 26d). While these experiments relied solely on shape and location information to determine the region of the individual shutters, the addition of color or intensity information enabling the exclusion of dissimilar regions would obviously make the strategy more robust.

To eliminate incorrectly labeled shutters, a shutter pair model (Figure 26e) guides the construction and application of a heuristic measure for pairing shutters. The heuristic measure rejects a shutter pair when two shutters from the set of initial shutters are not within half the height of each other, are further apart than four times their average width, are closer than one-and-a-half times their width, possess different brightness, and have centroids that are not roughly at the same height in the picture. Further, these measures are combined in a scoring function which is applied to each pair of shutters, and that pairing with the best score is selected (Figure 26f). Note that this initial pairing is done using image-based features and two-dimensional geometry.

Figure 27 shows the final labeling for the shutters. There are three errors that we wish to point out. First, the pair of shutters on the end of the second story of the house is missed completely. At first glance, this omission can be attributed to the fact that none of its regions was picked up in the initial set of regions. Perhaps this is a failure of the first step of the interpretation strategy or is due to the lack of an additional strategy. The question is: How much evidence is there for a pair of shutters? That the system missed this pair does not seem to be too reprehensible in light of there being an extremely narrow region (only one or two pixels wide) for one of the shutters while the other is obscured by the tree. A possible solution to



Figure 27. Final Results of Labeling for Shutters

This figure shows the regions that correspond to the shutter-pairs established in STM.

this problem would be to have a second shutters strategy (of the type described in the final section) which depends on having a complete instance of the model of the house. Then we could work from the supposition that a window *could* be there and look for shutters, accepting them on less evidence.

The second type of error occurred where one shutter of a pair was found but the shutter pair itself was missing (Figure 28). In this example, the left-most shutter pair on the front wall of the house (under the lower eave of the roof) and the hair on the back of the person's head are exactly the same color; thus, since there is no local evidence for separation of these two regions, we would expect that the segmentation system would be unable to separate them. Despite the fact that the shutter is not clearly segmented, and is therefore not initially detected, there is significant evidence for the shutter pair: the color and intensity of the region is almost identical to that of the unmatched initial shutter, portions of the shapes are

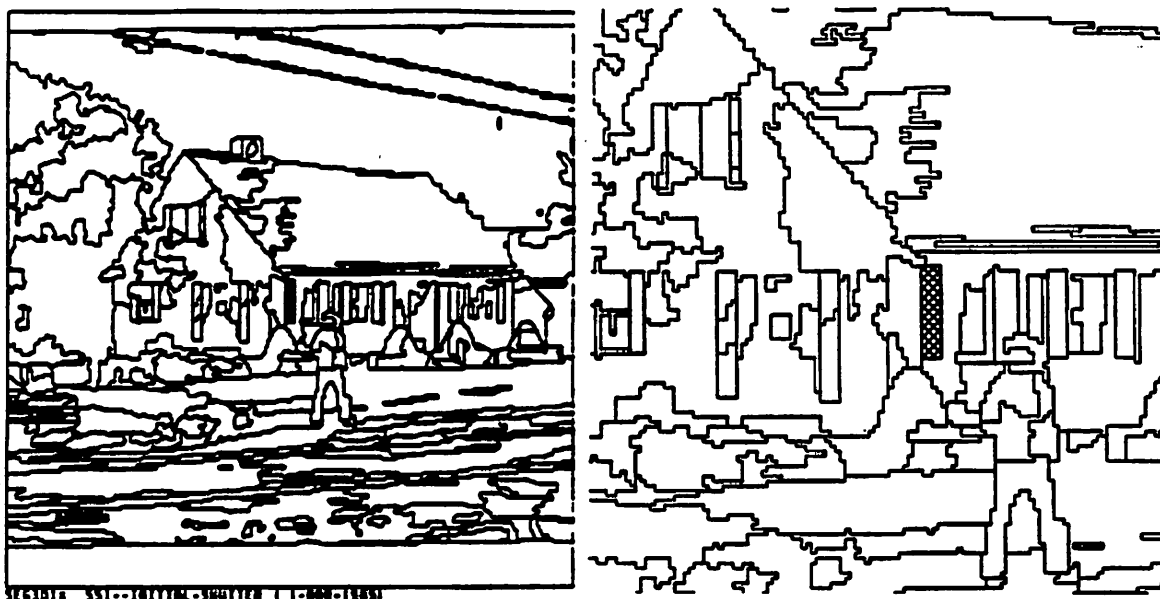


Figure 28. Initial Shutter Without Shutter Pair Hypothesis
 When a shutter hypothesis does not participate in a shutter-pair hypothesis additional evidence can be sought in the image for the “missing” shutter of the pair. An object-instance specific model for the shutter pair, derived from other shutter pairs on the same wall or in the same house, can be used to guide the search for that evidence.

similar and there are portions of the larger region that match the shutter model. Since the initial shape measurement used to test for rectangularity is based on the shape of the whole region, one approach to this problem would be to make this test of rectangularity more flexible and general, allowing for partial matching. In addition, a closer interaction between the segmentation and interpretation processes might permit a region such as this to be resegmented based on the need for a rectangle to complete the shutter pair. Finally, this type of error suggests that a “clean-up” strategy could combine the evidence of the one shutter of the pair that *was* found with the support of the descriptions of the other shutter pairs in the house interpretation to suggest where to look for the missing element of the pair. As in

this case, when another shutter pair in the interpretation had shutters of similar size, that shutter pair could be used as a template to find the "missing" shutter of the target pair.

The last problem is that the left shutter pair on the end wall is partially occluded. Thus in the labeling, and in the projection of that shutter pair on the house model instance (see following discussion), they are described in the interpretation as being different sizes. One solution to a problem like this is to have a mechanism that explains the unexpected difference in size. In this case, where the explanation is occlusion, the missing part of the smaller shutter would be added to make both shutters the same size. It would also be true that the smaller shutter would be made larger in the case where such a problem had been caused by segmentation errors. As one possible solution to the situation in which the two shutters in a pair are described as being different sizes we propose this heuristic, one that does not require the complex explanation mechanism: increase the size of the smaller shutter in the shutter pair to match that of the larger one. This would be applied when there was sufficient evidence that there should be a shutter pair; thus, for example, when the two merged regions were similar in color and the smaller region matched the shape of part of the larger region, the smaller region would be assumed to be a shutter. Heuristics such as this should only be applied when there is sufficient information to verify their results. When, for example, other shutter pairs have been interpreted and when the house geometry is available, the newly hypothesized shutter pair can be checked for size and placement consistency (e.g., it should be nearly the same size as other shutter pairs).

These problems are typical of the types that are discovered during the development of an interpretation system once it is working. Some of these problems can be overcome with minor adjustments in interpretation strategies. By using different

features for measurements or otherwise altering the decision criteria, an interpretation strategy can frequently be shifted to include a previously excluded case without altering previous results. However, this type of adaptation can be taken only so far; at some point, an additional, more general interpretation strategy is needed.

In step three of the strategy, the "posting" step, each shutter-pair hypothesis is added to STM as the description of an object with two related parts, the shutters of the pair. In this step the interpretation is made available to other schemas, and the shutters schema now initiates interaction with other schemas. This is a particular case of a general principle which we used in developing interpretation strategies, which is that interactions with other schema instances should be deferred as long as possible.

After posting the initial set of shutter-pair hypotheses, in step four, the shutters interpretation strategy waits for the house-wall hypothesis to be posted, at which point the house-wall and shutters schemas interact. While the house-wall schema interpretation strategy has a step for adding each shutter-pair to the description of the appropriate house-wall (as parts), the shutter schema interpretation strategy completes the description of the shutters as objects on the surface of the house-wall, using what three-dimensional information is available in the house walls hypothesis. What results from this interaction is that each shutter-pair is assigned to the spatially appropriate house-wall, and when the three-dimensional position and orientation of a house-wall are known, the shutters are given a three-dimensional position and orientation.

For each shutter-pair the precise placement of the shutter in space is computed by taking the centroid of the shutter region (in the image) and determining where that point would lie on the corresponding surface of the house. The height and width

of the shutter in the image plane are also projected to the surface of the house. Then the description of the shutter is constructed by computing the position of its four corner points on the surface of the house face, such that the shutter has a height and width that would produce the regions seen in the image (see Figure 29). This produces a description of the shutter in three dimensions, which can now be used to verify the spatial relations of the shutters within the pair and to correct small errors in placement and size estimate caused by the earlier two-dimensional interpretation.

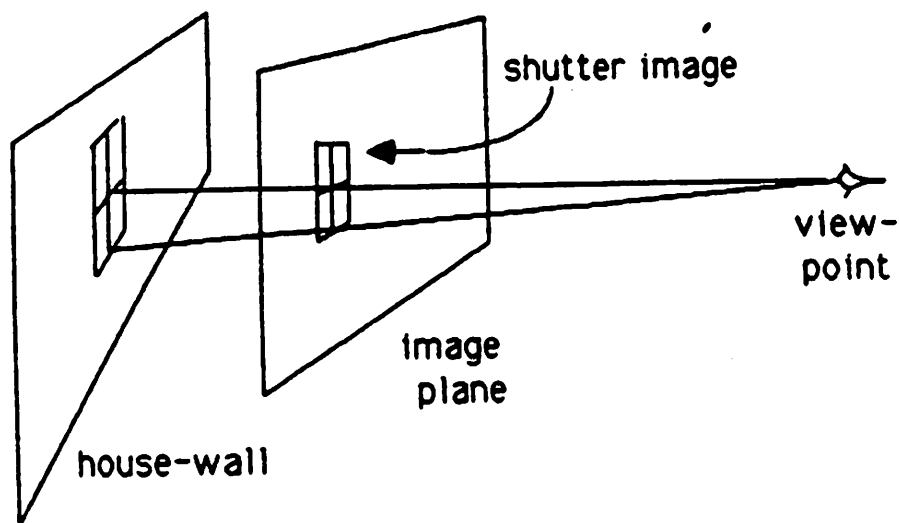


Figure 29. Projection of Shutter Image to House Surface

When both the image position of the shutter and the three-dimensional position and orientation of the house wall are available, the three-dimensional position and orientation of the shutter (on the house wall) can be hypothesized.

3.2 Road - Exemplar and Geometric Constraints

The implementation of the interpretation strategy for the road schema follows from a description of a road as having straight sides and a surface that is fairly uniform in color. To start, we use texture and color measures in a scoring function of the type used in the grass schema. The road schema actually embodies a description of two types of roads: dirt roads or light-colored roads (called "light-road") and

asphalt roads or dark-colored roads (called "dark-road") – as specified in Figure 30. The first step in the interpretation strategy for road is to run two scoring rules, one for each type of road. For each of the rules the region with the highest non-veto score is picked as an exemplar. When either rule returns an exemplar, it is used as the seed region for a road. If both rules return a candidate, then a new schema instance is formed and the interpretation strategy starts an additional instance, splitting the flow of control so that one instance is associated with each seed region.

The scoring functions for the two types of roads are not very general. In addition to being restricted to a limited set of viewpoints (e.g., roads with their centerlines projected to cross the image horizontally), the color features were developed from a small sample set. The goal was to develop an additional schema, not necessarily a good one, but one that was sufficient to produce interpretation results, so that the experiments with the entire system would involve several schemas simultaneously active. This interpretation strategy is one of those that was less well developed.

There are several possible approaches to recognizing multiple objects of essentially the same type that differ in only a few aspects. One method is to have the interpretation strategy discover all the objects of that type; this is the approach taken by the shutters schema. Another approach is to have the interpretation strategy find one object of the given type; this is the approach taken by most of the schemas discussed. Methods that lie between these extremes are also possible; for example, an interpretation strategy could keep a list of the possible candidates and return the most likely one first, waiting for a request-to-continue to return the next. Alternatively, the interpretation strategy could create goals for additional occurrences of the object with the new information about the previous hypotheses, causing new schema instances for the same schema.

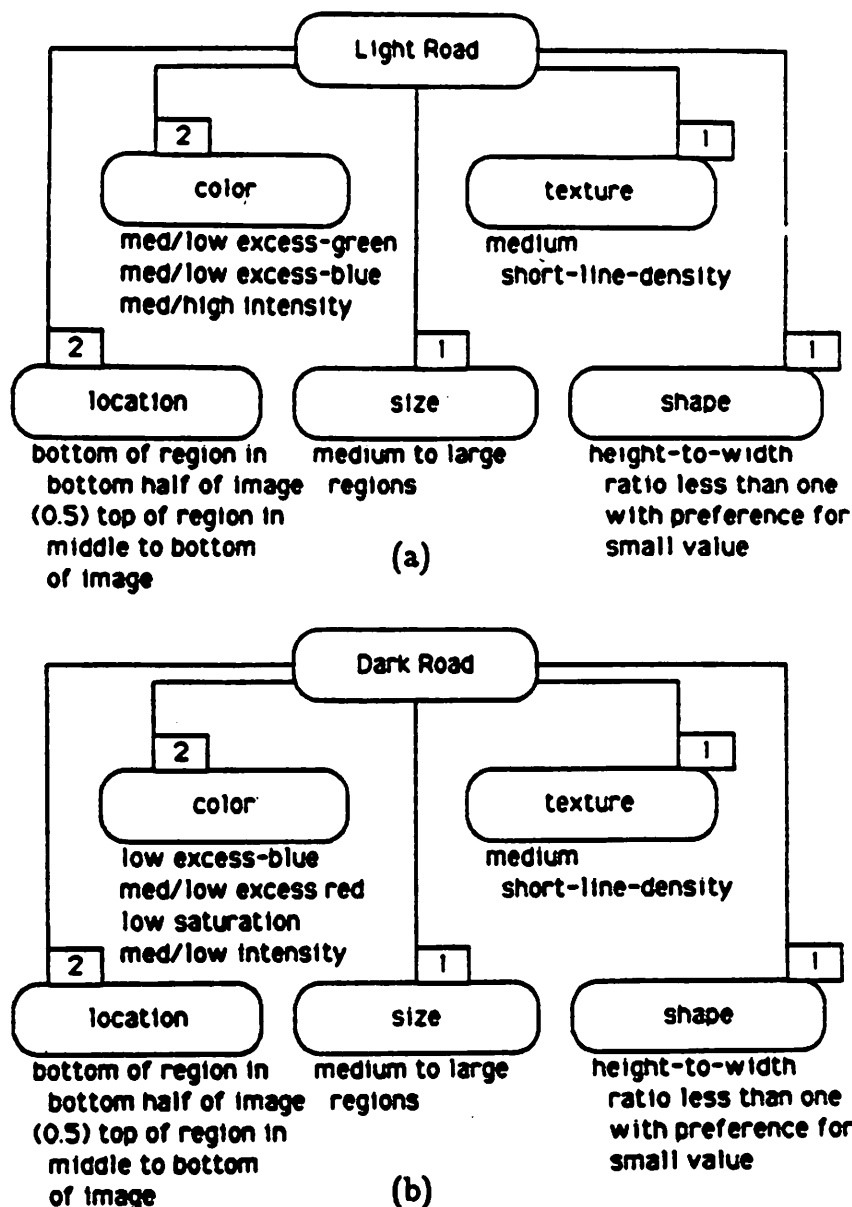


Figure 30. Combination Scores for Road

Features used by the two scoring functions for a lateral or transverse road model. (a) Light or dirt road and (b) dark or paved road are variations of the same model. Note that, although depicted separately here, they are expressed by the same basic description with varying parameter values. The identical parameter values are not duplicated in LTM. The (0.5) indicates relative weight, all others being one, for weighted averaging. Some features such as the shape measures are viewpoint dependent and based on the specific (transverse) model of a road.

The road schema was an experiment in handling the problems that resulted when a schema discovered more information than was necessary to satisfy the goal which caused its activation. In this case, it is possible for the schema to discover two different types of roads: a dirt road and a paved road. In some house-scene images the road surface is connected and of one type; however, in some images there are two different road surfaces. This occurs, for example, when a dirt driveway and a paved road are both visible. Of course, these two types of roads could be treated as different types of objects altogether, but then we would be unable to take advantage of the features common to both types. In some important sense these two types of road are really variations of a common, basic type of object, a road.

The road interpretation strategy provides an opportunity to experiment with another approach. Because there are two strategies (for dark and light road) which differ only in the parameters used, and because the scenes we were interpreting had a road of only one type, we attempted to respond to the case of multiple objects by having the interpretation strategy create a new schema instance to satisfy the same goal. Thus, the schema instance executing this interpretation strategy creates a copy of itself and the two instances pursue separate interpretations. This effectively splits the control of one process into that of two schema instances.

Due to our lack of sophistication in our method for handling the relation between goal requests and the schema instances serving them, the idea of having the strategy split control did not prove generally useful. What is to be done when two instances respond to a goal for one object? Of course, with road (as an object) the problem is more complicated because each road identified by a separate schema instance could, in fact, be part of a single road. These problems were complicated by the mechanisms available for communication. It was not clear how to communicate the effect of the splitting of the process to other schema instances. In practice, minimal communication was achieved by setting local variables in each

schema instance to indicate that there was another schema instance working on the same problem and by permitting the schema instance with the larger image area to respond to the goal. The hypothesis generated by the schema instance with the smaller image area is only posted in STM (where it is subsequently added to the description of the ground-plane by the ground-plane schema instance). The communication necessitated by this arrangement was handled in an ad hoc manner because it fell outside the scope of the originally designed goal-and-contract based communication channel. It is clear from this need that more general strategies will require other types of communication than those based on goals and the schema instances satisfying them.

In the second step, common to both strategies, the exemplar region is extended by a heuristic application of the road model in a routine that searches for additional road regions (see Figure 31a). The exemplar region is assumed to be in the area of the road. Thus, some parts of the road that extend along the length of the road, e.g., center lines or curbs, will be manifest in lines in the image that intersect the exemplar region but extend past its boundaries. In the second step of this strategy we extend the road hypothesis by selecting from the database of straight lines extracted from the image (discussed in Chapter 2) those long lines which pass through the exemplar region and testing all the regions through which those lines pass for an extension of the road. The extension regions are selected by measuring a difference in color and texture from that of the exemplar region and selecting those regions having a difference value below an empirically determined, fixed threshold. The selected regions are added to the description of road, and that description is then posted in STM as a hypothesis for road (see Figure 31b).

This strategy is based solely on the image features associated with road; no three-dimensional knowledge is used. For scenes requiring a more complete interpretation of road, an additional strategy could be developed based on knowledge



Figure 31. Results for Road Interpretation Strategy
 (a) The initial region and (b) the extension produced by the road interpretation strategy.

of the three-dimensional relations between the parts of the road. Roads are essentially continuous surfaces, which are smooth with locally parallel sides; in addition, they are frequently straight. There are often some markings in the center that are parallel to the sides. Also, the surface of the road is roughly in the same plane as that of the ground-plane. These facts, and others derived from a three-dimensional model of the road, could be used to make a more general road strategy.

In the strategies for both the shutter and road schemas we have seen examples of how the geometry of the object (in the object description in the schema) is used to extend a region or refine a set of regions into a more complete description of the object, encompassing more of the image and including more detail in the description.

3.3 Roof (First Strategy) - Exemplar and Geometric Construction

The roof schema contains two interpretation strategies; we will present the first here and the second will be covered in the next section. The two principal steps of the first strategy are essentially the same as for shutters and road. Many of the images of roofs have a similar color and texture. Thus, an application of the feature-based scoring function can be used to pick out regions of that type. Unfortunately, not all the roof regions can be obtained in this way, so we used a modification of the exemplar selection and extension strategy. In the first step of the strategy the scoring function outlined in Figure 32 selects an exemplar region based on color (medium to dark grey) and texture (moderate).

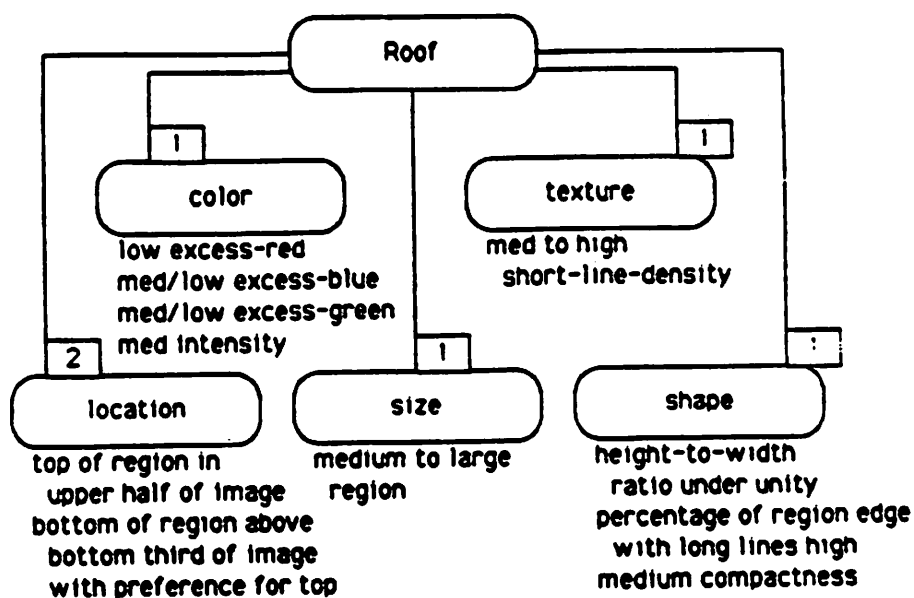


Figure 32. Combination Score for Roof
This figure shows the features of the scoring function for roof.



Figure 33. Initial Roof Region and Line Data
(a) Shows the region selected as the roof exemplar and (b) the lines from the image data base.

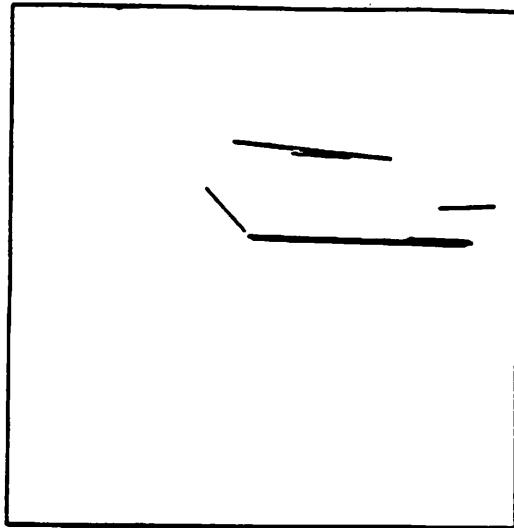
As with the shutters and the road, we encounter fragmentation; the roof region shown in Figure 33a is fragmented. We again use the knowledge of the object shape to extend the hypothesis region. As with the road, some object parts that are expected to be linear will show up in the image as straight lines. Thus we can start with an exemplar region, select related lines, and use these to find related regions. To extend the roof hypothesis, we take advantage of the knowledge about the geometry of the roof. The roof is known to be bounded by straight lines. Thus we look for lines on the boundary of the exemplar region (see Figure 33b). At the places where those lines extend beyond the region we can look for matching regions by using a color comparison.

This roof strategy is one example of using complementary types of data, in this case regions and lines from different low-level processes, to extend the interpretation. The algorithms for extracting the regions and lines make errors in differing ways. For example, a straight edge boundary omitted in the region segmentation is frequently reflected as a straight line. In addition, some object features are more naturally manifest in lines (such as edges) while others are more naturally sought after as regions (such as surface features). This stage of the roof strategy is an illustration of how these two sources of information from the image can be combined to achieve a more complete interpretation.

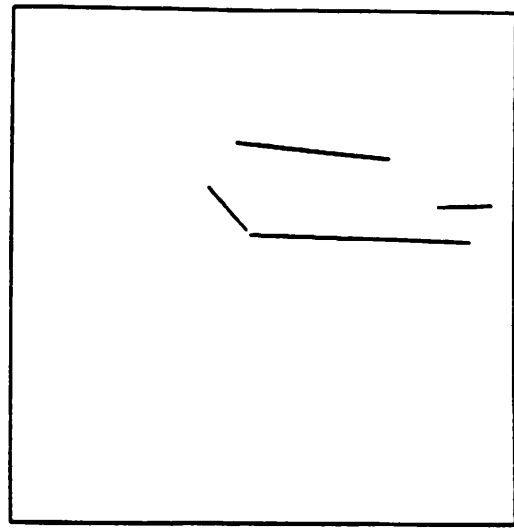
The second step of this strategy extends the initial hypothesis (the exemplar region shown in Figure 33a) by finding the boundary lines that are long relative to the size of the selected region – specifically, those lines which intersect the exemplar region boundary and have a length of greater than one-third the square root of the area of the region. These lines are selected from the set of high contrast long lines shown in Figure 33b. They can be thought of as evidence for edges that extend past the boundary of the initial region (see Figure 34a). Since many of these lines represent the individual edges of linear parts (e.g., the several edges of the gutter) nearly parallel sets of lines are represented by a single line. The longest line of each set of nearly parallel lines is selected, eliminating smaller lines that lie on or near the same edge of the roof (Figure 34b). Regions adjacent to these selected lines, having color and texture similar to the initial hypothesis region, are picked as extensions to that region (see Figure 34c). These regions are added one at a time, starting with the region that is most similar, until a rough test for a roof boundary is passed. The completion test checks for fragments of two sets of roughly parallel lines that could complete a quadrilateral and assigns a score based on whether the lines were found and their degree of fragmentation. This score is compared to a fixed threshold. Then boundary lines for these regions are partitioned into

Figure 34. Roof Interpretation Strategy

These figures show intermediate results from the steps of the roof interpretation strategy. (a) Long lines bounding the exemplar region. (b) Lines remaining after elimination of shorter redundant lines, used to extend the exemplar region resulting in (c) the additional region selected for extension, based on adjacency to the lines and similarity in color and texture. The extension is verified based on the existence of two sets of roughly parallel lines and the degree of fragmentation among those lines. This completes the first major step in the process. (d) The set of lines from the boundaries on all selected roof regions. (e) Redundant lines are removed and each colinear group of lines is replaced with a single line fit. (f) The candidates selected as those lines which best match a parallelogram. (g) The completed outline for the quadrilateral which will match the projection of a rectangle. (h) Comparison with all the regions from the original segmentation which have centroids interior to the projection of the roof's rectangular surface.



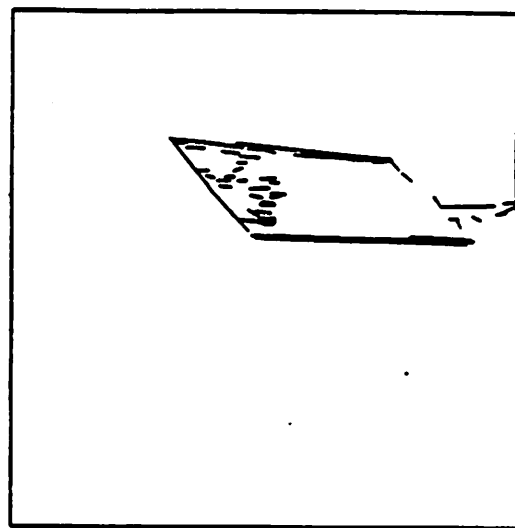
(a)



(b)

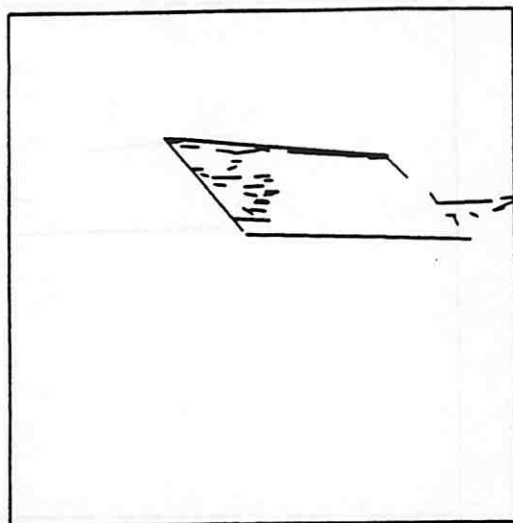


(c)

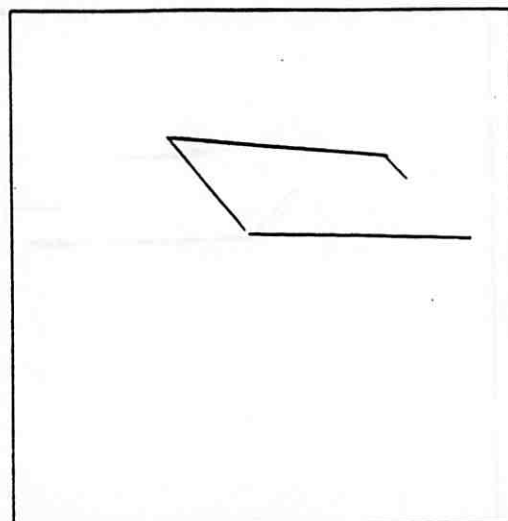


(d)

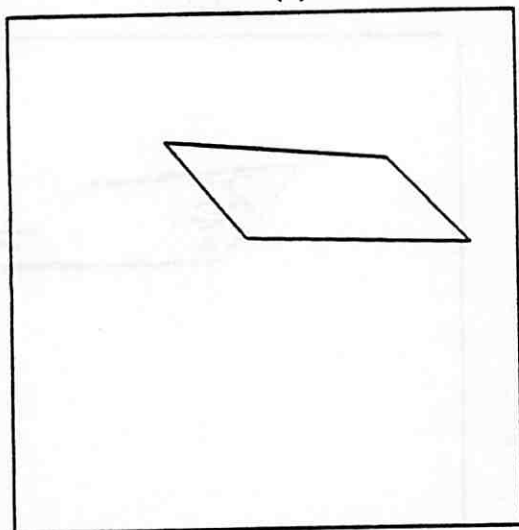
Figure 34.



(e)



(f)



(g)



543010 507--ROOF-EXTENSION (30-MAY-1964)

(h)

Figure 34 (continued).

colinear sets of lines (Figure 34d) and a line is fitted to each set (Figure 34e). A heuristic grouping function selects the longest lines that roughly fit a parallelogram (Figure 34f). These lines are assumed to correspond to the boundary of the roof area and they are extended to their nearest points of intersection (Figure 34g). This completes the roof boundary, which can be used to indicate the image area of the roof. Also, the segmentation can be labeled by selecting all the regions with centroids interior to the boundary polygon; this relabeling is only a rough approximation to the regions interior to the roof (Figure 34h). For example, the region on the right side of Figure 34h extends past the true boundary. This figure only shows a labeling based on the original segmentation. The true interior region is obtained by projection of the three-dimensional description into the image plane (see below). The lines and the region of the roof area are the image data items associated with the geometric description of the roof.

Once the image area and boundary lines associated with the roof are part of the description, knowledge of the geometric structure of the roof is used to guide the construction of the three-dimensional description of the roof. Knowing that the roof boundaries are the projections of the sides of a rectangle permits the determination of the surface orientation of the roof and the determination of the roof corners up to a scaling factor that represents the relation between size and distance inherent in a perspective projection. With the addition of an assumption about the approximate distance to the nearest point of the roof, the distance to each corner can be computed. Thus, we can derive the placement of the roof in space. The assumption about distance merely provides a scaling value that can be updated as additional information becomes available.

This interpretation strategy has a verification phase in the second step: a program measures the degree of correspondence of the description of the roof in STM with the description of the roof in the schema. It checks for correct number and

placement of the parts of the roof, in this case the edges of the roof and, also, checks to see that the edges are in the correct relation measuring the degree to which there are straight lines in the image. These measures are summed and normalized to form a verification score for the object. An additional verification, when the three-dimensional description of the roof is available, is to check that the surface normal computed for the surface of the roof is within reasonable bounds.

After the description of the roof is constructed, in the third step, nodes are added in STM representing the roof and its parts. Because of the availability of a geometric description of the structure of the roof, hypotheses can be created for the non-visible edges and surfaces. The occluded surface of the roof is positioned based on an assumption that the roof is symmetric about a plane perpendicular to the ground plane and passing through the crown of the roof. This information is used to construct a description of a three-dimensional instance of the roof as a pair of rectangular surfaces in space. Since the extended hypothesis for roof contains the full, three-dimensional description of the roof, the geometric description can be projected into the image to provide precise location information or can be used in further geometric constructions. In addition, the description provides clues for the subsequent hypotheses for the house walls. Because the roof and walls are parts of the house, such information is passed through the house schema. The position of the roof can be used by the house schema instance and, subsequently, by wall schema instances in forming hypotheses about the position and existence of the walls. This is recounted in detail in the section below discussing the house and wall interpretation strategies.

4. Key Feature Matching and Geometric Construction

In many cases, there are key image events which indicate that an interpretation should be attempted. Consider, for example, an image with a large blue region in the upper half. This event, alone, is enough to indicate that an interpretation as

an outdoor scene is quite possible and therefore could be attempted. Of course, once activated, the schema for outdoor scene may fail to produce an interpretation. However, in the case where there is sufficient evidence, an interpretation can be made. For situations like this, there is a need for interpretation strategies that can start from a key event.

A trivial interpretation strategy that is initiated by a key feature or image event is one which checks for the event and upon finding it present, continues with a more complex strategy. This is essentially the method used by the interpretation system in monitoring for data derived activation of the schemas.

There is, however, another situation in which key features provide a useful starting point for an interpretation strategy. When no other clear starting point can be found, a single image event may be enough to initiate an interpretation. The second roof interpretation strategy, discussed here, is an example of this type of strategy. It is based on the general heuristic that if no other interpretation paths for a roof (and its related house) can be completed, the activated schema should attempt to use evidence of a more tenuous nature to initiate the interpretation.

When the failure of the first roof strategy is detected by the verification program, one alternative is to invoke a second interpretation strategy. This strategy looks for a particular set of key features as an indication that the object might be present; in the case of the roof, we used a pair of long horizontal lines below the sky as the key feature. If such a pair of lines appears in the image, meets the criterion of being above the horizon and is the right distance apart (the distance threshold being a ratio of the length of the longer of the parallel lines), then the initial hypothesis is made for the roof being between those two parallel lines. At this point the strategy proceeds, as did the first strategy, by attempting to complete the roof description based on its geometric properties.

The particular set of features that we chose is only an illustration of many that could have been used for roof. Other are easily imagined: for example, a three-sided junction of the right shape, or perhaps opposing corners that are complementary.

One interesting class of key features is the type based on context. For example, a roof might be indicated by a long horizontal line below the sky or a long horizontal line above the shutters (the height of the shutter could even be used to predict where the bottom edge of the roof was expected). Strategies based on context in this way are discussed in Section 6.

This roof strategy is an example of how the trade-off between hypothesis generation and verification can be used to advantage. When the strategy based on a generation method that selects only a few initial hypotheses fails, we turn to a strategy that selects regions with less care and relies more heavily on the verification program to keep incorrect hypotheses from being put into STM.

5. Key Parts and Using the Composition Hierarchy

We now turn to the discussion of other interpretation strategies which, like the second roof strategy, are based on the use of key features; specifically, we will discuss interpretation strategies that use the part-whole (or composition) relations from the schema network. In these strategies, interpretation depends on the prior recognition of certain key parts. For example, the house schema contains an implementation of this strategy that requests the interpretation of the roof and the walls of the house (as key parts) before it proceeds to form a description of the house. When a hypothesis for either of these two parts is returned, the subsequent interpretation is built on that part; and verification in each step is accomplished by checking that the parts found are in the correct spatial relations (when there are no multiple parts, there is no verification). Thus, interpretation of key parts of the object is used to drive the interpretation of the whole object. The first step of this strategy, that

of forming a tentative hypothesis, is the acquisition of the key part or key parts. The second stage, the construction of a description of the object, usually combines additional parts resulting in a description that is posted in STM. From that point, the interpretation strategy might continue to monitor STM for the posting of other parts and include them in the description when they are posted. Schemas employing this type of part-whole strategy are those for ground plane, outdoor scene, house scene, house, and walls.

5.1 Ground Plane - Collection from STM

As a first example of a strategy that uses part-whole relationships, we will look at the ground plane interpretation strategy. This strategy starts by issuing a goal request for grass which is the key part. When a grass hypothesis is put into STM by the responding grass schema instance, it is also returned to the ground-plane interpretation strategy which uses it to construct the initial hypothesis. Because grass is a part of the ground plane, each region associated with the grass hypothesis should lie on that plane. Ideally the position of such regions would be determined by back-projecting the regions from the image to the ground plane using a known camera model. However, the position and orientation of the ground plane are not known, the surface of the ground is rarely a plane, and there are already small errors in position due to segmentation problems (among other factors); thus, an approximation will suffice. We start with two simplifying assumptions: that the ground is a horizontal plane and that the distance computed using the centroid of the region is representative of the distance of the region (see Figure 35). This projection of the centroid was used for expediency; it would be possible to project the region to the hypothesized ground plane and use that area on a surface in space. In fact, additional types of projections are also possible. For example, with objects known to stand on the ground plane the region could be projected to a flat surface parallel to the image plane and perpendicular to the ground plane. Additionally, the regions could be projected to predefined shapes, e.g., rounded cones for bushes. Each of

these projections implies a set of assumptions about object shape, relative position, and relative attitude. We only implemented one such set projection. Specifically, by assuming that the camera is upright, exactly parallel to the ground plane, and at a known height off the ground, then the distance to an arbitrary region is given by

$$d = h \frac{f}{r}$$

where

h = height of camera off ground plane

f = focal length of camera

r = distance from center of region to horizon in the image

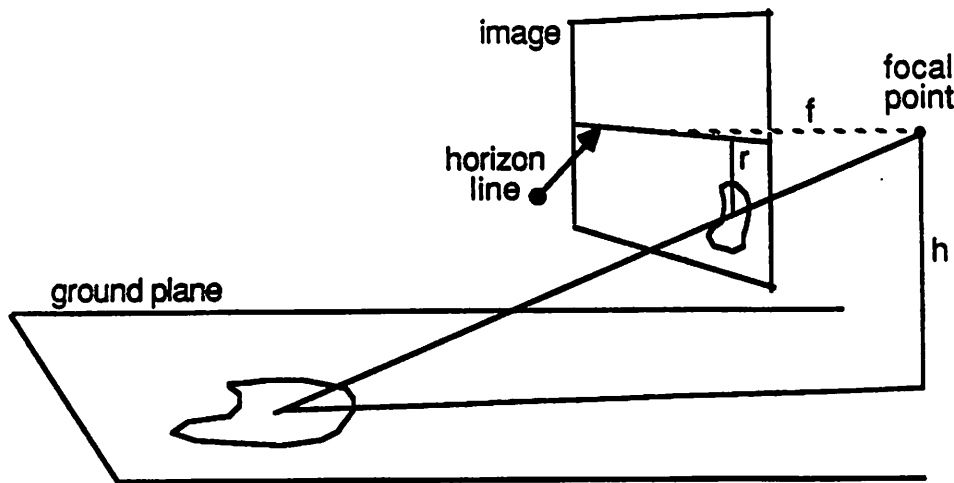


Figure 35. A Camera Model Gives an Approximate Distance
Simplifying assumptions about the position of the ground plane, and the position of the regions of objects that lie (or stand) on the ground plane (grass, in this case) allows the computation of the approximate distance for the object surface area represented by the region in the image.

After the ground-plane hypothesis is augmented with the distance estimates, it is posted in STM. In order to add the relations for objects on the ground plane that are not central to the interpretation, the strategy continues after the posting stage with a fourth step that monitors STM for additional related objects. By waiting for

these hypotheses to be posted this strategy can verify whether the objects appear to lie on the ground plane. Several heuristics can be used to check whether an object can be hypothesized to be lying on or sitting upright on the ground plane. If the object is of a type that should be upright (e.g., shrubs), the distance estimate given above can be checked with other distance information for consistency. Such objects may also have a distinctly flat (i.e., horizontal) bottom extent; when present, this is additional evidence. Objects with known geometric structure (e.g., the house) provide additional evidence that can be checked for consistency. If an object thought to lie upon the ground plane has lines that provide more than one distinct vanishing point, its surface orientation can be compared with that of the ground plane. When such objects do appear to be on the ground plane, a relation to that effect is added between the ground-plane hypothesis and the hypothesis for that object.

A possible alteration to this strategy would be to have the ground plane description modified to reflect the additional information gained from objects thought to be on the ground plane. Those objects with a known spatial relation to the ground plane (from independent knowledge or assumptions) could be used to confirm the position of the ground plane or to adjust the position parameters when they were not in agreement with the placement of the object. For example, the feet of a walking person would either be on some other object or on the ground plane. When a portion of the image is interpreted as a person, the ground plane hypothesis could be updated to reflect those relations. Other objects thought to be on the ground plane – because of their proximity to the ground plane and the likelihood that they would, in fact, appear on or be touching the ground plane – could also cause an adjustment in that hypothesis. To make such adjustments, additional strategies would have to be designed to track the changes so that the relations with objects originally thought to lie on the ground plane would not be made inconsistent as the ground plane description shifted.

It also seems reasonable that the ground "plane" should be described by some other geometric model than that of a plane. Being able to represent local deformations would simplify the task of aligning the bottoms of objects with the ground plane without disturbing previous interpretations. Further, such a description is truer to the facts: curved slopes, small hills, and gentle dips are ubiquitous in "level" ground. However, this problem is rather complex and will require much more sophisticated mechanisms.

5.2 Outdoor and House Scenes - Basic Part-Whole

In addition to the grass schema, both the outdoor scene and house scene schemas use strategies based on finding key-parts and pursuing the additional part-whole relations stored in LTM. Each of these strategies is an instance of the general strategy of attempting to satisfy all the LTM relations of an object. The variations come from the relative importance assigned to the relations and are manifest in the order in which the relations are satisfied. The most important parts, called key parts, are selected by knowing which will most likely lead to the identification of the object, and these parts are sought first. In the case of the outdoor scene, the key parts are the sky and the ground plane; in the case of the house scene the key part is the house. When the key parts have been interpreted - that is, when hypotheses for the key parts have been posted in STM and returned in response to the goal requests - the interpretation strategies for these schemas construct an initial hypothesis. Then they request hypotheses for other parts and add them to the description when they are found. Thus, the description of foliage gets added to the outdoor scene description and the description of road gets added to the house scene description. In addition, the description of the house scene is also added to the outdoor scene, because house scene is a sub-class of outdoor scene.

5.3 House and Walls - Key Parts and Using Geometry

The interpretation strategy for the house is a more complex strategy based on part-whole relations. It resembles that of the roof schema in that it relies on geometric information and the relation of that information to the placement of lines in the image. In addition, the strategy of the house schema relies on specific information about the parts of the house and the "best" order in which to attempt to recognize them. There are two interrelated interpretation strategies for house, each based on the recognition of a key part: one for roof and one for walls. The roof-based interpretation strategy is attempted first, because when the roof can be recognized, much more information can be obtained about the details of the house geometry. Specifically, the identification of the roof permits the construction of the house geometry description, whereas the identification of the walls does not.

Basically, the house is assumed to be a simple geometric polyhedron consisting of a roof and walls. As illustrated in Figure 36, the walls and roof are polygons that share common edges. The model does not specify the actual shape and size of the polygons; rather, a routine that constructs a description of the house in the image expresses knowledge about relations among the planes and edges in the model. Relations used by that routine are the expected angles between the surfaces, the expected shape of the surfaces, and limits on the height, width, depth, and pitch of the roof. Default values are provided for each parameter when no image-based evidence is available. The following assumptions are made about the geometric structure of a house: the two roof surfaces are rectangles of identical size; the front and back faces are rectangles of the same size; the left and right end faces are pentagons of the same size; each end face is symmetric about a vertical axis; the front and back faces meet the end faces at right angles; and the front, back, and end faces are vertical (see Figure 37).

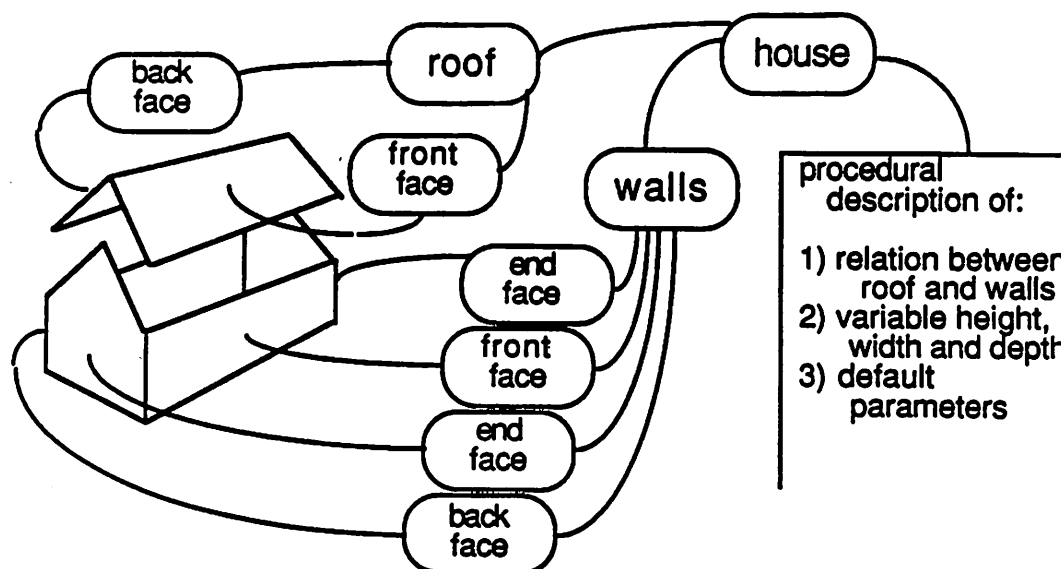


Figure 36. Model of House Geometry

The model of the house geometry, including the shape of the wall and roof surfaces modeled as simple polygons. Associated with these are the illustrated attachment relations, e.g., the front wall is attached to both side walls. In addition, the wall surfaces are grouped, the roof surfaces are grouped, and the roof is attached to the walls. Additional relations among the parts of the roof and walls are also represented. For example, the end walls are symmetric about a vertical line, the angle of the peak of the end walls is the same as the angle between the roof surfaces, and the walls are vertical. See Figure 37 for additional relations in the roof. Also, there are both absolute and relative default values for height and width.

In the first strategy, when a goal request for roof causes a response of a roof hypothesis, that hypothesis is used to construct an instance of the house model (see Figure 38). By using the surface normal and position information from the roof description we can derive an instance of the wire-frame model of the house. The surface normal, when projected into the horizontal plane (assuming that the house is upright), produces the normal to the front face. The normal to the other visible face is derived from the cross product of the two normals, one from the roof and one from the front wall. The cross product produces the correct normal, because

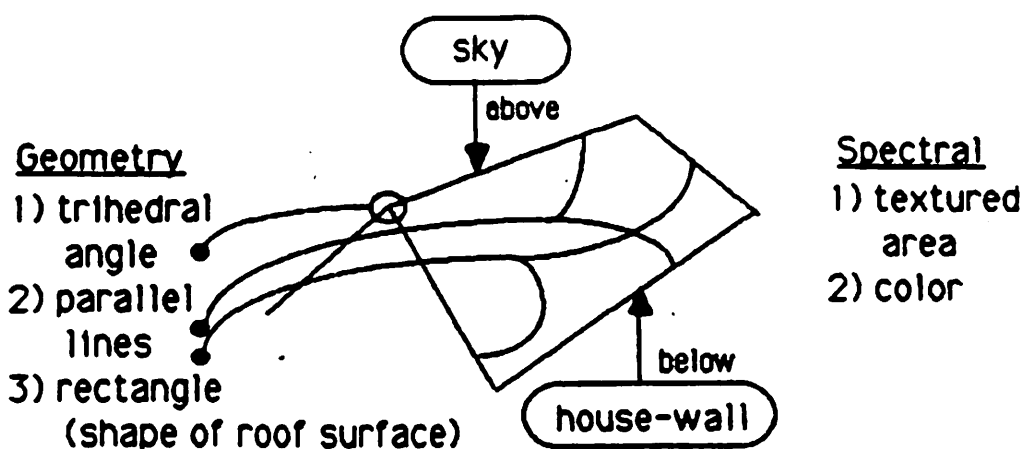


Figure 37. Relations in Roof Model

The roof model specifies that the two roof rectangles are of identical shape and size. Additional geometric relations are included. For example, the trihedral angle shown joins the peak of the roof and the peak of the side wall of the house, and the roof surfaces form an angle that is bisected by a surface perpendicular to the ground plane. Several of the relations implicit in the rectangular shape of the roof surface are explicitly recorded. The roof is bounded by two sets of parallel lines and its corners are right angles. Finally, the relations between the parts of the roof and other objects are represented. The roof is below the sky, above the house wall, and (by transitivity) above the ground plane. The color and texture of the roof surface are inferentially represented by the exemplar selection scoring function.

the front face and the roof are both at right angles to the end surface of the house. When more than one surface of the house is visible, the normals to all the visible surfaces of the house together with the image position of the vertices on the visible polygons enables us to compute the relative distance to all the visible points in the object. In fact, the absolute distances, except for a scaling factor s , can be computed. In a coordinate system with the origin at the focal point of the camera, each point P in the object has coordinates in space that are given by:

$$P = sP'$$

where

s is the scaling factor

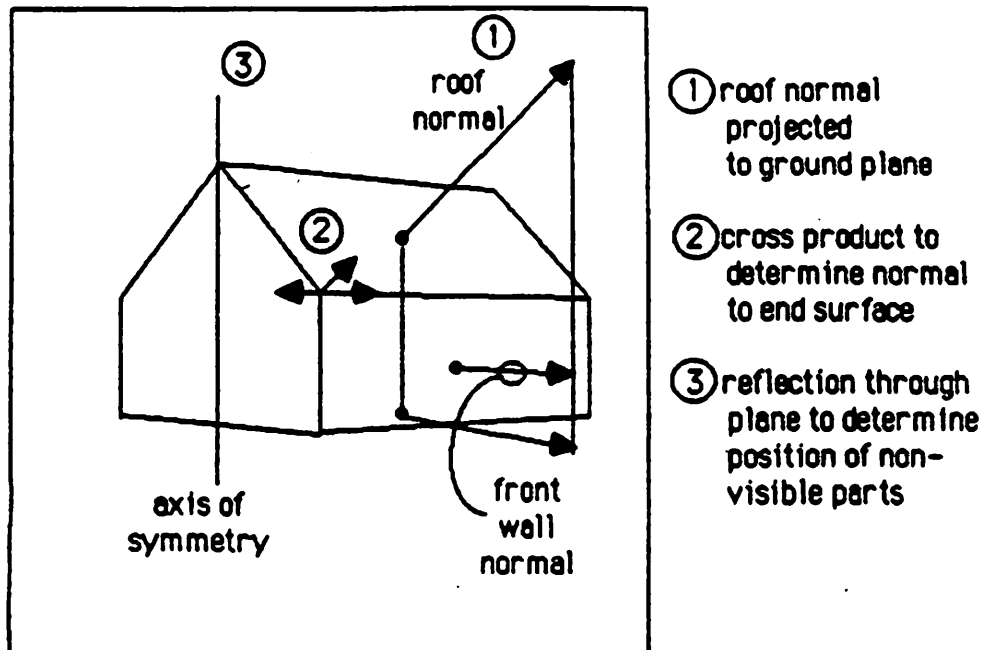


Figure 38. The House Geometry from the Roof
 The geometric description of the roof can be used to initialize the geometric description of the house. See text for description.

P' is the object point at an arbitrary scale

The set of points denoted by the point P' are those three-dimensional points of the original interpretation. This scaling function captures the idea that non-geometric information (e.g., knowledge about typical house sizes) must be used to distinguish between a small house that is close to the viewer and a large house that is farther away.

We verify the determined position of the points of the house by using expected values of the height of the house. If these expectations are violated, the scale factor can be adjusted to admit the possible interpretation of the house as being at a different distance than originally estimated. In addition the strategy uses heuristic information from the house schema to estimate where to expect the bottom of the

house. In order to approximate the height of the house, an estimate of the size of one story of the house is computed from the height and pitch of the roof. This value is used to compute the relative default value for the height (see Figure 37). For the gable roof model, the height of one story is assumed to be approximately that of the vertical distance from the horizontal plane going through the eaves of the roof to the peak of the roof. The default height of the house is two stories (one story plus the height of the roof); however, this default is adjusted when the relation between the house and the ground plane is known. Then, the house is assumed to be that height, in integer multiples of one story, such that the bottom of the house is close to the ground plane.

When the description of the house is put into STM, many other schemas can anchor their interpretations to this description. For example, the description of three-dimensional geometric structure provides very tight constraints on the locations within which to search for the line segments in the image corresponding to the edges of the walls. Thus, the walls schema strategy limits its examination of the image to those areas which are likely to contain relevant edges. This is illustrated in Figure 39, where we see that image line segments supporting the hypothesized model are within a couple of pixels, and frequently directly on, the projection of the hypothesized house boundary.

When the roof schema instance reports failure and no roof hypothesis is available, the second strategy requests a walls hypothesis. An initial house hypothesis is constructed if a walls hypothesis is returned. In this case the walls hypothesis depends solely on being able to locate image features for window parts (e.g., the shutters) and is, therefore, more prone to failure. In the case that a walls hypothesis is returned, it does not contain the type of geometric information available from the roof schema (e.g., the walls interpretation strategy does not provide a geometric description); thus, this type of house hypothesis is less complete. However, the walls

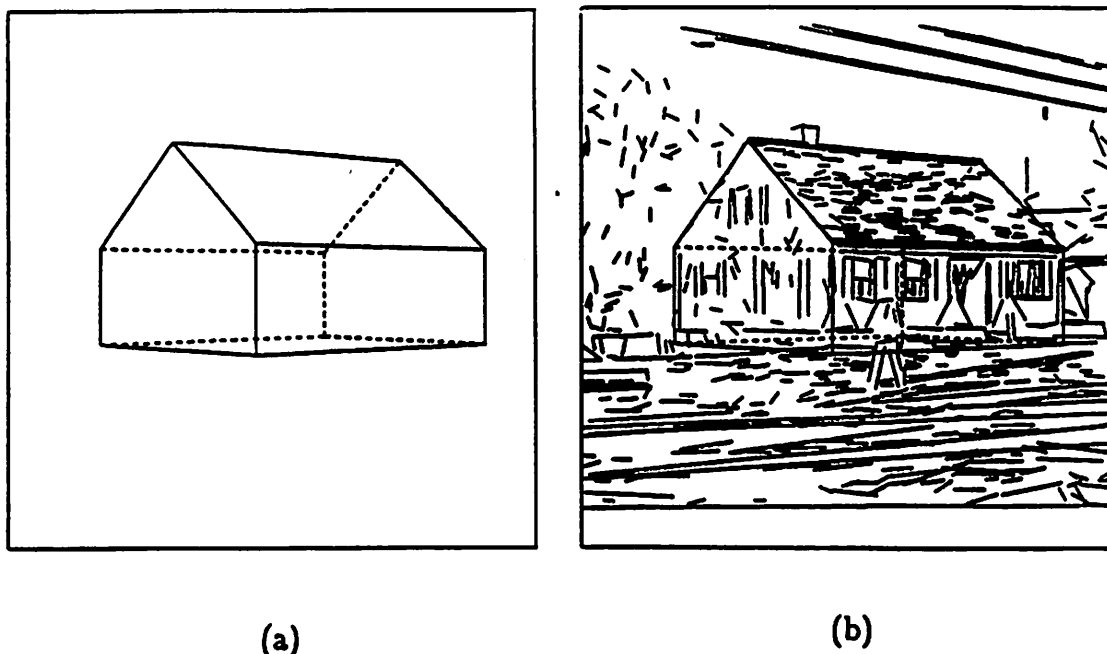


Figure 39. The House-Wall Hypothesis

(a) An image projection of the house instance derived from the roof instance and the house geometry. There is a close match between the data and the instance of the house model derived from the roof. (b) The geometric description of the house superimposed on the lines for the image. This house instance can be used to identify the position of each particular wall of the house.

hypothesis can provide the image area of the walls as discussed below and this can be used to constrain the image position of the model instance. In addition, if there is enough evidence to determine which view of the house appears in the image, the appropriate two-dimensional to three-dimensional relations can be used to aid in the construction of a model instance in which the default values fill in unspecified parameter slots.

The walls interpretation strategy is also based on key parts. To initiate the interpretation for this schema, the strategy requests shutters. If a shutters hypothesis is returned (and posted), the walls are described as those image areas adjacent to

the shutters that contrast most greatly with them. The shutters hypothesis is the hypothesis for the set of initial shutter pairs.

In the second step of the walls interpretation strategy, the walls are refined using the house geometry. When the interpretation strategy in the house schema adds the geometric description of a house to the house hypothesis this addition is posted in STM. The interpretation strategy in the walls schema (waiting for that posting) then determines from the geometric information the precise boundaries of the walls in space. With the addition of the geometric information, hypotheses for each separate wall can be formed, differentiating the image areas associated with the house walls into areas for distinct walls. The shutters can also be assigned to a particular wall, completing these details for the house model instance.

If no shutters hypothesis is returned, then the first step of the walls interpretation strategy fails. In this case the interpretation strategy for walls relies on the posting of the house geometry hypothesis to STM. If the house geometry hypothesis can be constructed, then the walls geometry hypothesis can be constructed without the supporting image information of the walls areas. Ideally, an extension such as this would require some other independent path of verification, and thus could rely on the verification procedures to confirm the walls hypothesis. In the current system the unverified walls hypothesis is all that is available in this case.

In the interaction between the second house strategy and the walls strategy there is a potential for deadlock. The house strategy waits for walls hypothesis to be posted in STM. Moreover, in the case that there are no shutters, the walls strategy cannot construct a hypothesis until the house geometry is posted; when it encounters this situation it waits for the posting of the house geometry. In short, if neither shutters nor roof are hypothesized then these two schema instances could both be waiting on one another. To avoid this problem, in the case that no

shutters have been found, the walls strategy responds with failure after the first step. It then waits for a request to continue. This allows the house schema to control the interaction. The house schema requests the resumption of the walls schema instance only when finding the roof of the house permits the construction of the house geometry hypothesis. This particular interaction illustrates how the communication between schema instances can be used in critical control situations.

The interpretation strategies for the house and walls schemas illustrates how hypotheses for key parts can be extended using knowledge about the geometric structure of the object. In the process of constructing the hypothesis for the house, the interpretation strategy extracts data confirming the three-dimensional structure.

6. Context-Initiated Interpretation

The final type of interpretation strategy is motivated by the fact that once part of the scene is interpreted, many types of objects are much easier to recognize. Of course, the other schemas discussed in this chapter rely partially on contextual information for their strategies; for example, the interpretation of the house-scene is made easier by the prior interpretation of the outdoor-scene, and interpretation of the house depends on interpretation of either roof or walls. The strategies discussed in this section, however, depend entirely on this type of contextual information. For example, the interpretation of the lines in the sky, shown in Figure 40a, is restricted to one of a small number of things by the context established when the surrounding area is interpreted as sky. To distinguish among the possible choices requires that the interpretation strategy check for only a few clues; for example, the choice between a wire and a jet-trail could be made based on the contrast between the "line in the sky" and the sky. Thus, the sky represents a "context" for the interpretation of a "wire-in-the-sky." One strategy for interpretation of the wire-in-the-sky is to assume the sky hypothesis is correct and to base the interpretation

of the wire on that assumption. For the purpose of illustration, we designed two strategies of this type: one for "wires in the sky" and the other for "telephone pole."

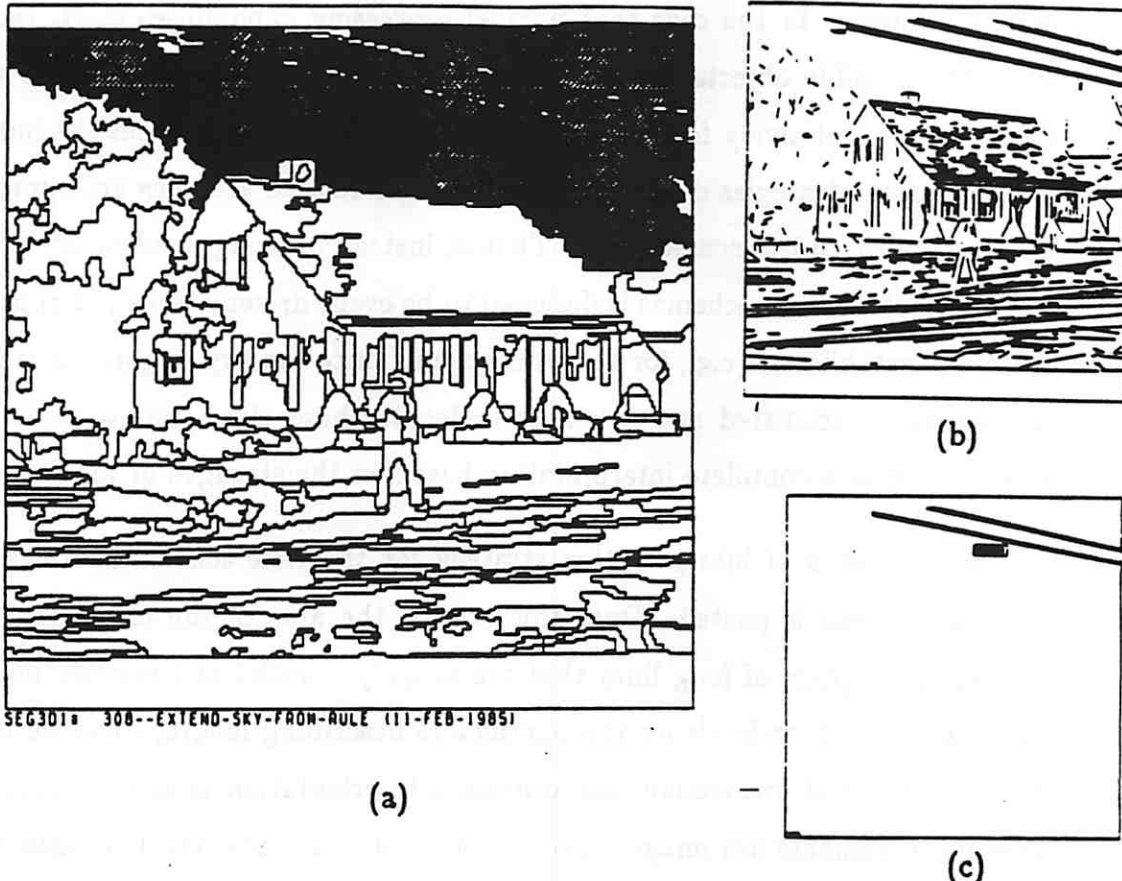


Figure 40. Context-Based Wire Interpretation Strategy

The interpretation strategy for wire assumes a prior interpretation of sky. (a) The initial sky region. (b) The lines in the database from which the strategy selects long and parallel lines based on the context indicated by the sky hypothesis. (c) The final interpretation of the wires in the sky approximated as long narrow quadrilateral.

In developing these types of strategies, there is a trade-off between having the strategy of each schema test for its corresponding object and having an umbrella schema (called, perhaps, objects-in-the-sky) categorize the unclassified features in the sky and select those schemas that should be activated. Which of these two

control styles to adopt would depend upon the potential number of objects indicated by the context, upon the amount of processing needed to determine if a particular schema were appropriate, and upon the availability of parallel processing capabilities. In the case that parallel processing capabilities exist, especially when the possible objects are few in number and the tests to distinguish a given object require relatively few resources, then one alternative is to let the individual interpretation strategies determine whether their related schemas are appropriate. Such an approach is discussed here. That is, instead of being invoked by any global schema, each of these schemas is designed to be event driven. Once the appropriate context is established (e.g., for the wire schema, once the sky hypothesis is posted) the schema is activated and attempts to locate those clues that will permit the construction of a complete interpretation based on the strength of the context.

The first step of interpretation strategy for the wire schema is to wait until a sky hypothesis is posted. Once this occurs, the area of the image labeled sky is checked for pairs of long lines that are roughly parallel and roughly horizontal (determined by thresholds on the parameters describing length, distance between lines, difference of orientation, and difference in orientation to the horizontal). All these measurements are image-based. Figure 40b illustrates the lines selected in a test image. If such lines are present, they are extended to form the longest set of parallel lines which can be supported by the data, and the interior area is checked for an indication that it is darker than the surrounding sky. If these verification checks are met, a wire hypothesis is formed and added to STM (see Figure 40c).

It is interesting to note that this interpretation strategy is greatly simplified by using the assumption that the context exists. The number and complexity of tests and measurements needed is reduced, a point that is reinforced by our second example.

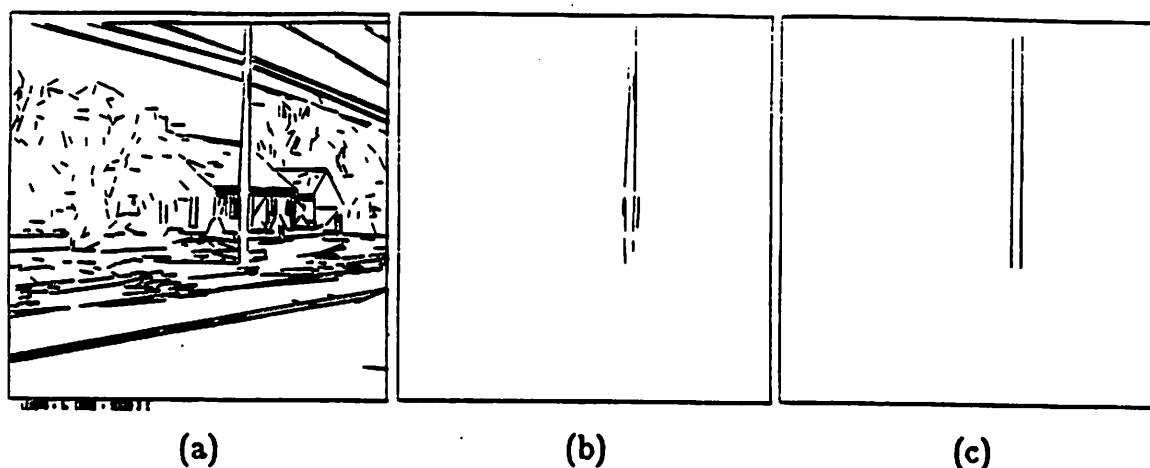


Figure 41. Telephone Pole Interpretation Strategy
(a) Telephone pole in image, (b) intermediate results, and (c) final results of interpretation strategy for telephone pole.

The telephone-pole strategy depends on a less specific context than the wire schema. It expresses the idea that any object in an outdoor scene not interpreted as part of a man-made structure (a house in our example) and having a pair of vertical parallel lines extending across the horizon line becomes a candidate for a telephone pole. Figure 41 illustrates the steps in this strategy. When the pair of parallel lines is discovered, it is first extended and a hypothesis image area is formed. As a final verification step this area is checked to see that the regions are low to medium intensity, roughly the same hue and roughly the same intensity; although a wide tolerance on the intensity differences among regions is permitted because there is a high ratio of mixed pixels within those regions.

Many clues exist for additional verification and extension of the interpretation. The wires in the sky touching the upper part of the telephone pole region are a very strong verification clue. The size-distance relation of the telephone pole is available from its relation to the ground plane. The occlusion implied by the broken roof line

for the house gives an additional clue as to relative distance. Finally, the placement next to the road is an additional clue.

These two strategies were designed after the experiments shown in the next chapter were run, hence they are not included in those interpretations. However, they are straightforward and illustrate how simple interpretation strategies can be used to add detail to an initial interpretation that supplies strong context.

7. Using Interpretation Strategies – Summary

In this chapter we have explored the details of the interpretation strategies of the schemas for the current interpretation system. In an ongoing interpretation, these interpretation strategies provide control by selecting actions, directing communications, and constructing the corresponding hypotheses of each schema. Each interpretation strategy is patterned from a general template strategy based on a hypothesize-and-verify cycle. These diverse strategies all start with a tentative indication of the presence of the object in the scene and accumulate data and related hypotheses in a description of the object which is eventually added to Short Term Memory (STM).

The accumulation of these partial interpretations results in the construction of a network describing the scene in the image. With the addition of contributions from each interpretation strategy, the network describing the scene grows to cover the objects in the scene and to account for some of the detail. Having looked at the individual interpretation strategies and their roles in the schemas, we discuss in the next chapter the effect of the overall action of the schema network by examining the results of the interpretation process.

The complexity of the general interpretation task is easy to see from the diverse and varied types of knowledge used in these interpretation strategies. Additionally, it is clear from this discussion that these strategies would have to be made far more

robust to work in a general interpretation setting, because of the wide range of detailed knowledge required by interpretation. While it would be ideal to be able to describe objects and the methods necessary to recognize them in some uniform declarative representation that a system with a general control structure and inference mechanism could then apply to interpretation, the development of many of these strategies is more naturally explored by constructing programs to perform interpretation. Such an approach fosters an experimental exploration of the types of knowledge and procedures necessary for interpretation, similar to that experienced in the development of expert systems. By probing the more complex interpretation problems and accumulating an understanding of the types of information and control required, we are approaching the point where much more of this knowledge could be expressed in a general declarative framework. These early strategies only point the way to future development of more general and sophisticated strategies.

CHAPTER IV

SCHEMA CONTROLLED INTERPRETATION

In this chapter we discuss the control of interpretation by interacting schemas. This method relies on the representation of fine-grained knowledge in the form of interpretation strategies. Interpretation is accomplished by the selective application of interpretation strategies within the contexts of increasing specificity that arise from partial interpretations.

Methods for control and the selection of knowledge have been discussed throughout the first three chapters of this dissertation. In the review in Chapter 1 we introduced several systems for image interpretation and provided a review of several major control methods used by those systems. In Chapter 2 we examined a general framework for knowledge representation and the control of interacting schemas; and in Chapter 3 we showed how specific interpretation strategies controlled the selective application of fine-grained knowledge to an interpretation task.

In this chapter we show how the communication mechanisms introduced in Chapter 2 are applied to control interpretation through the interactions of several schema instances.* After a brief introduction in which we review the motivation for using schemas as a basis for interpretation, Section 1 describes our experimental system and Section 2 gives a detailed description of a goal-directed image interpretation. An example of a data-initiated interpretation is the subject of Section 3. Following that, Section 4 is a summary of additional results, selected to highlight additional features of schema instance interaction. Finally, Section 5 is an overall summary.

* We continue our practice, started in Chapter 2, of referring to schema instances as *instances*, as distinct from object hypotheses and verified object hypotheses.

Several issues relevant to control of image interpretation processes and our schema-based approach, though previously introduced, are listed again:

- In the interpretation of an image, multiple interacting sources of information are available.
- Much of the processing is undertaken with some degree of uncertainty.
- Some hypotheses are more easily verified than others.
- Both goal-driven and data-driven processing strategies have practical value.
- The representation of large amounts of detailed knowledge is necessary for image interpretation.
- The interactions among some "chunks" of knowledge can be quite complex.

These issues are interrelated and, taken together, form a partial set of "specifications" for control in an image interpretation system.

By considering the vast difference between the type of input that an interpretation system receives (an image) and the type of output it would produce (e.g., a symbolic description) one can infer the need for multiple interacting sources of information. There are many possible ways, let us call them grouping processes, in which image events or descriptions can be organized into more abstract structures. Furthermore, new structures are produced by selectively attending to some parts of the image information and not to others. Thus, each new type of structure potentially requires the representation of new knowledge. In addition, grouping processes rely on the results of other grouping processes; for example, the corners in the image may indicate a polygon (the first grouping process) and the features of that polygon may serve as input to a process that identifies patterns of polygons. There are hierarchial relations among the grouping processes, and the intermediate structures which they form are additional sources of information. In addition, information can also come from sources other than the image and processes acting on it: models of expected objects and scenes, relations between possible events in the image, and inferences to be made from evidence gathered.

Control-related information provides further sources of knowledge that must be integrated in order to effectively and efficiently interpret an image. For example, interpretation strategies can use knowledge about which types of hypotheses will be more easily verified and information about which processes to invoke next. A major issue related to control is the ubiquity of local uncertainty. Each sensor and each of the grouping processes is a potential source of error. Image interpretation can be characterized as a control problem when one observes that a globally accurate interpretation can be produced from the contributions of many locally errorful and uncertain processes. Control mechanisms must combine complementary types of information and take advantage of redundancy to reduce the likelihood of error. However, even error-free sensory data and processing would still leave us with the problem of ambiguity. Only in rare cases does a single piece of evidence indicate a given object type. Any single image feature will usually be an indication of several possible world events. This is also true of many of the structures produced by the intermediate grouping processes. Further, any world event is inferred not only from the observation of several image features or intermediate structures, but also from the relationships among these observations. Thus, from the observation of a specific fact there is great uncertainty about the implied interpretation.

One approach to overcoming the problems of uncertainty is to utilize the rich dependencies and redundancies by starting with those hypotheses that are most easily verified and inferring from them the contextual foundations necessary to support hypotheses which are more difficult to verify. This method of expanding interpretations outward from the most easily verified hypotheses provides a "focus of attention" mechanism in which processing is expended in those areas of the interpretation for which evidence is already accumulated. It does not deal, fully, with the issues of managing uncertainty; however, it has proven to be a (mostly) adequate approach for our experiments.

1. An Experimental Schema System

The experiments described in this chapter illustrate how a collection of schemas in a schema network can control interpretation. We have developed a schema network to interpret suburban house scenes based on the interpretation strategies described in Chapter 3 (see Figure 42). The control information for interpretation is distributed within the interpretation strategies of the schemas in the schema network. For these experiments, we developed a set of guidelines which are related to the methods of communication and techniques for schema activation introduced in Chapter 2. They were used to determine how control information should be distributed among schemas, and their derivation resulted from considering:

- which object parts should be represented by a separate schema and which should be recognized within the schema for the whole object,
- whether schemas for object parts or other related entities should be activated in series or in parallel,
- whether an interpretation strategy should wait for the results related to a requested goal (causing the interpretation for that object to be suspended) or to proceed with further processing and either wait later or periodically check STM for the results,
- where and how to test for conflict in interpretation, and
- how to react to interpretation failures.

1.1 Considerations in Implementation of Control

Each schema represents an object, but not every object is represented by a schema; for example, the individual shutter of a shutter pair is only represented as part of the shutter pair within the shutters schema. In constructing a schema network, we had to choose whether to describe object parts in the schema for the object or as relations between that object and another object represented by another schema. In part, this choice must be based on the level of conceptual resolution required by the interpretation task. Thus, while the boundaries for the roof do not each have a separate schema and are described in the roof schema as parts of the roof, the roof itself is represented by a schema even though it is part of the house.

Figure 42. Schema Network

This schema network was used to control all the interpretations discussed in this chapter. The graph includes both schema nodes (rounded corners) and nodes for implicitly represented objects (square corners). The implicitly represented objects are part of the interpretation strategy of their associated schema node. For example, the knowledge of roof geometry is implicit. Thus, the relations among the edges of the roof rectangle in space, the relations among the angles and points in the plane of the roof, and the possible spatial relations among the lines matching edges of the roof in the image, as well as other spatial and geometric relations, are part of the roof interpretation strategy. In addition, two classes of relations are represented: hierarchical and spatial. The hierarchical relations are represented with the arrows labeled with a circle. They are "part-of" for the compositional hierarchy, and "kind-of" for the specialization hierarchy. The spatial relations are represented by arrows labeled with a square; they include the relations below (above), standing-on, flat-on, and adjacent. Many of the spatial relations are exclusively or redundantly represented within the interpretation strategies of the schema.

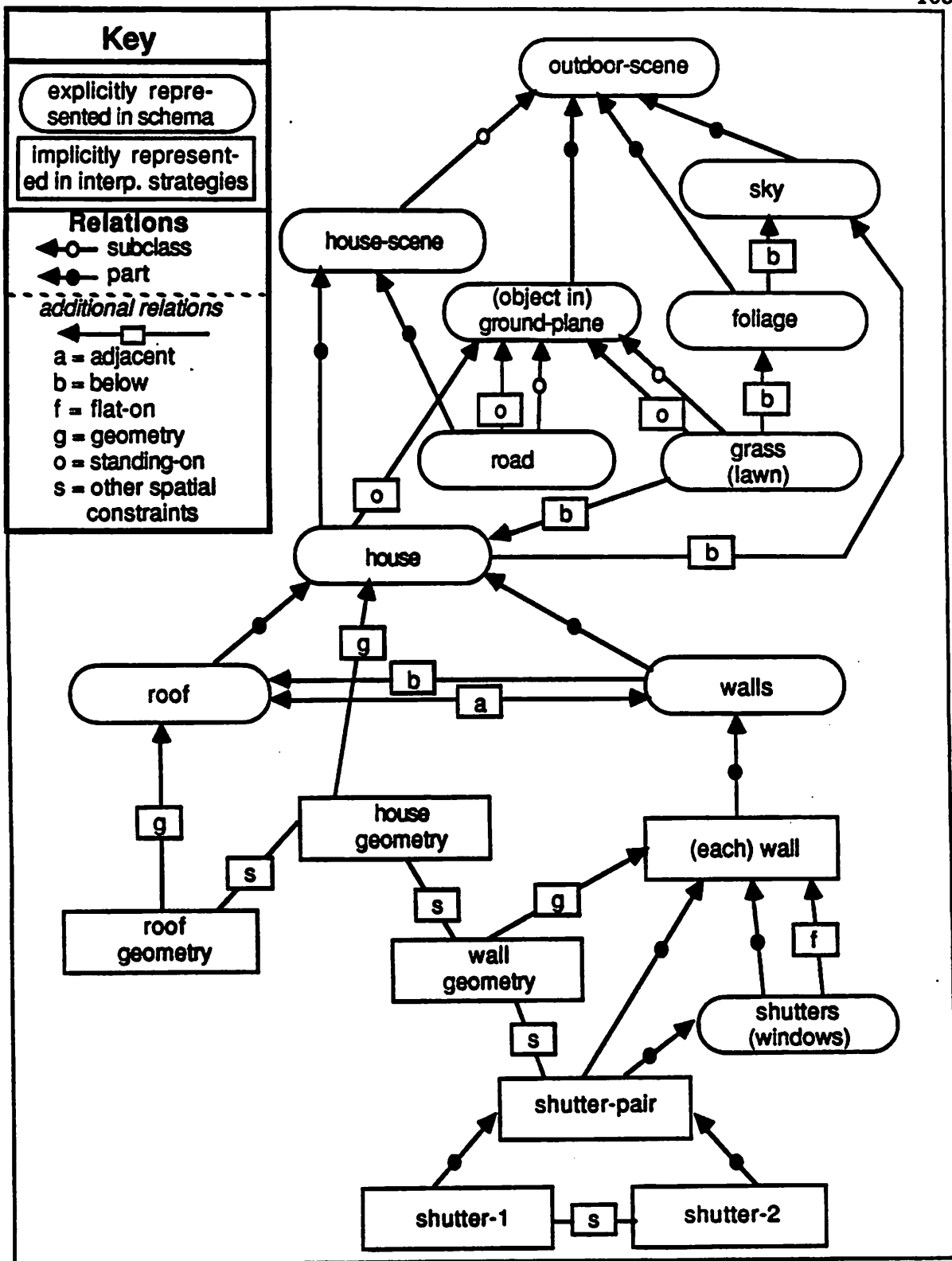


Figure 42.

Generally, we chose to represent an object with an independent schema based on the availability of an interpretation strategy for that object. As an illustration, there is a strategy to recognize house wall regions (e.g., walls), but each individual house wall (e.g., the front wall) is recognized by part of the walls interpretation strategy in an interaction between that strategy and the strategy of the house schema through a house geometry hypothesis. Thus, the front wall has no separate schema.

Another decision was whether to request goals for object parts in series, waiting for the response to each request before making the next request, or to request the goals in parallel, making all the requests at once and then waiting for each request to be satisfied. Clearly, when interpretation strategies are independent, with little or no interaction with other results, it is best to make requests for object parts in parallel. This is especially true when the interpretation strategies involved have roughly the same likelihood of contributing to the ongoing interpretation and are roughly of the same cost in terms of computing resources required. Thus, for example, the sky, foliage, and ground plane are all parts of the outdoor scene and they are requested in parallel.

When the interpretation strategy can not proceed without a particular part, especially if the subsequent interpretation requires a substantial amount of processing resources, then its execution is delayed until the successful completion of the interpretation for the key part, and an alternate path is followed if that key part is not found. As an alternative to requesting a goal, particularly when the related hypotheses are not essential, a polling of STM can be used in which the strategy is written to either periodically check for the occurrence of the hypothesis in STM or wait for the creation of a required hypothesis in STM. Actively checking STM is chosen when there are other tasks which the strategy can perform without the requested hypothesis, such as refining interpretations through further data acquisition.

Finally, there are two general decisions related to failures in interpretation. The first, conflict resolution, occurs when two alternative interpretations are made of some portion of the image. We decided to place the conflict resolution processing in the lowest schema in the compositional (part-whole) hierarchy which had the conflicting schemas in common. For example, the interpretation of some regions of the image as foliage and as house wall is resolved by the outdoor scene schema, the site in the schema network where the conflict could be most easily detected.

The second issue related to failures in interpretation is how to react to schema strategy failure. Depending on the situation, we chose among the following alternatives: abandoning the interpretation attempt, retrying the current interpretation strategy with different parameters, or attempting an alternative interpretation strategy. When a schema has been activated in response to a goal, an abandoned interpretation is reported as a failure to the schema instance which requested the goal. Clearly, failure in an interpretation is an additional piece of information that can be used by the requesting interpretation strategy; however, our experience did not lead to any general and definitive observations on methods for reacting to interpretation failure.

1.2 Our Model of Parallel Computation

In the foregoing discussions of schemas, schema activation, and schema instances, we have stressed the potential for parallel computation of partial results. However, the implementation of the schema system for the experiments described in this chapter was done in LISP on a serial machine. Thus, before we discuss the interpretation results, we will close this section with a few short notes on the model of parallel computation which we used and its simulation.

In the most general setting, at any point in the interpretation there will be one or more active schema instances and each of these will be executing one or more interpretation strategies. Ideally, in the case that unlimited computing resources

are available, each schema instance would be assigned a processor upon activation. Likewise, the strategies within a schema could be assigned a processor each. Since the only interactions of schema instances are through communication by message passing and accessing the shared memory of STM, the interactions among schema instances are well defined. A similar delimiting of the interactions among interpretation strategies could be based on common variable names. Thus, between points at which interaction is required, the execution of interpretation strategies in separate schema instances can proceed independently.

We chose to simulate the parallel execution of the strategies in separate instances by interleaving the execution of the program steps associated with them. Since each schema instance contains the full program state of its associated strategies, it is a simple matter to interleave their execution. The simulation program first "switches contexts" by restoring the program state for the set of interpretation strategies associated with the schema instance. Then one or more program steps are selected and executed and, finally, the program state is saved in the schema instance. Thus, the simulation program can cycle through all the active schema instances executing program steps.

The method for determining the order in which to cycle through the schema instances could influence the outcome of interactions. In a truly parallel system, the outcome of unanticipated interactions is undetermined. To simulate this we felt it best to randomize the selection of which schema instance was next in the cycling. In this way, all other things being equal, each schema instance gets (on the average) an equal amount of the computing resources without any implied ordering of *unrelated* program steps within the simulation of parallelism. In this way we simulate the parallel activation of schema instances with unlimited processing resources.

In practice, computing resources are finite. Specifically, we are interested in the case where there are several processors, but we assume that during some part of the interpretation there will be fewer processors than active schema instances. Since the schema instances will have to compete for the computing resources, we find it necessary to derive some method of selecting schema instances and determining how much of the available computing resources they should receive. If we assume that each schema instance should have equal access to the computing resources, then our simple random selection method is adequate. However, we decided to rank the schema tasks for selection by an "activation score" based on some simple heuristics, representing (roughly) the relative "importance" of the instance to the ongoing interpretation.

The heuristics used to rank the schema instances are based on three ideas. The first heuristic is that schema instances which are effectively producing results should be allowed to continue to do so. Thus, if one schema instance is "doing better" than another during the computation it should get relatively more computing resources. Each interpretation strategy produces a score indicating how well things are going in the portion of the interpretation handled by that strategy; this score is used for the "fitness" part of the ranking. The second heuristic is that schema instances that are covering a larger portion of the image should be given more computing resources; they are (in some sense) more important to the overall interpretation. Each interpretation strategy records how much of the image is currently explained by its interpretation and the ratio of the size of that portion to the whole image is used to determine a "covering" score. Finally, the third heuristic is that every schema instance should "get a chance" at the available computing resources. To reflect this, the ratio of number of cycles used by the interpretation strategies to the total number of cycles is converted to an "idle cycles" score. These scores are combined (normalized and averaged) to produce a score for the schema instance.

In order to preserve the non-deterministic nature of the selection for execution of a program step we continue to use random selection of schema instances. However, reflecting the notion of competing schema instances, that selection is weighted so that (on the average) the schema instances get computing resources in proportion to their scores.

The simulation is written in LISP and the image processing algorithms are written in LISP and FORTRAN. Graph manipulations (in STM and LTM) were facilitated by a graph manipulation language, called GRASPER [(LOW78)], written in LISP. No effort was put into an efficient implementation: each interpretation described in this chapter takes on the order of 12 hours of CPU time! Rather, our goal was to examine interpretation strategies and communication methods; therefore, much of the cost in time was incurred to keep the system easily modifiable. Overall, our experience indicates that a straightforward reimplemention for efficiency, were this desirable, would achieve a speedup of a factor of about four, thereby reducing the time for an interpretation to somewhere around three hours of CPU. Additional speedup could be achieved by abandoning the flexible experimental environment and by optimizing control and communication software. Clearly, further research is needed into more effective software implementations and into methods of using parallel and special-purpose hardware.

2. An Example of a Goal-Directed Interpretation

We now turn to an examination of the results of some interpretations. Using the set of scenes introduced in Chapter 1 we will illustrate the interactions of the interpretation strategies described in Chapter 3. For the first example we will discuss in detail the sequence of observed schema activation for the goal-directed interpretation of the image shown in Figure 43.



Figure 43. Image for Goal-Directed Interpretation Example
This is a photograph of the scene used in the first interpretation example. Major portions of the scene, including the centrally located house, are interpreted by the application of the schema network.

Since this first example is intended to emphasize the goal-directed nature of activation, we initiate the interpretation activity by a user request (goal) for an outdoor scene interpretation. Recall from the description in Chapter 1 that the system has access to an initial database of intermediate representations for the image. These representations are initialized by a region segmentation and straight line extraction process, and procedures that extract a set of attributes of the regions and lines (shown in Figure 44). The user-supplied initial goal causes the interpretation of the image primarily through the creation of subsequent goals and the

Figure 44. Image Data for Example Interpretation

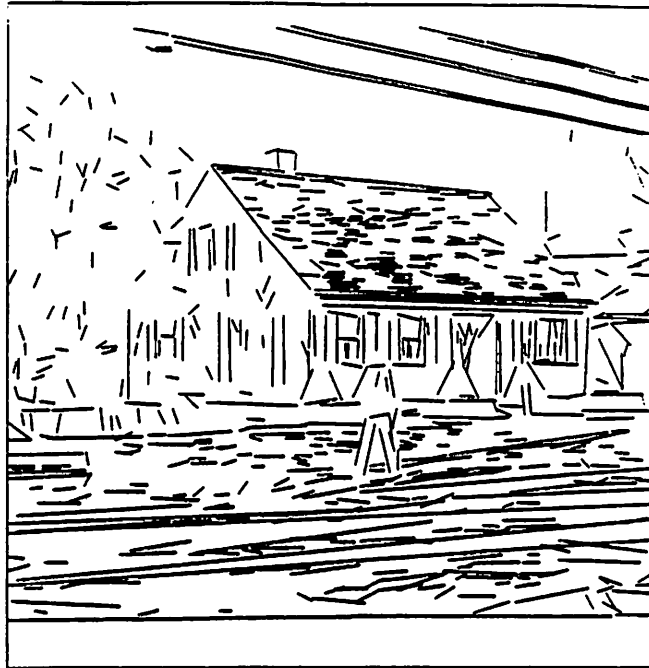
(a) The digitized image, (b) the set of straight lines, and (c) the region segmentation used to start this interpretation. Note that the digitized area includes part of the boundary of the original slide; this area is not included in the interpretation. The results from this interpretation example were also used in the illustrations of many of the interpretation strategies (Chapter 3).



(a)

Figure 44.

(b)



(c)

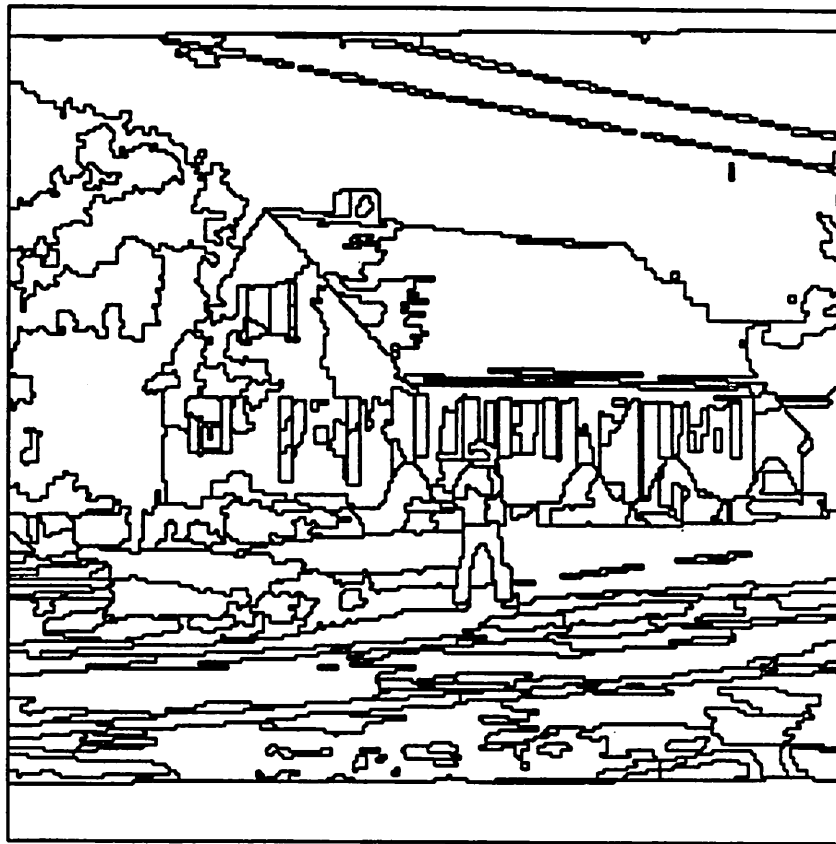


Figure 44 (continued).

interpretation strategies invoked to satisfy them. We will see, however, that not all the interactions between schemas in this interpretation are by means of goals. Since many parts of the image can be interpreted early (in parallel), this provides a basis for subsequent processing. Early hypotheses allow the system to be more selective in the application of specialized knowledge. For example, the system starts by identifying the sky regions which in turn are used in combination with other partial results to verify the outdoor-scene hypothesis, the basis for the rest of the interpretation.

Since Chapter 3 gives the details of how each schema constructs the description of a particular object, the discussion that follows will concentrate on the interactions between the schemas. The chief device used to convey this interaction is the trace of schema activation shown in Figure 45. The creation of a goal request is shown by a right-pointing arrow and the creation of a responding hypothesis is shown by a left-pointing arrow. (The horizontal scale and the length of the arrows is not significant.) Distance down from the top of the diagram shows the approximate passage of time; the basic time unit is the execution of one interpretation strategy step. Since the interpretations are carried out by a simulation of parallel activity, true concurrency did not occur; however, events at roughly the same position (vertically) are nearly concurrent. For example, the events labeled t_1 , t_2 , t_3 , and t_4 can be considered to be roughly concurrent, although t_2 must have followed t_1 .

In this example, the initial goal request by the user causes the activation of an outdoor-scene schema (see the point t_0 on the time line in Figure 45). Recall that in schema activation a schema instance is created and the interpretation strategy for the schema invoked. The interpretation strategy for the outdoor-scene schema requests goals for the subparts: sky, ground-plane, and foliage. These goals cause the creation of instances for their respective schema and the invocation of their

Figure 45. Time Trace of Schema Activation

This figure shows the time relations of goal requests, schema activations, and hypothesis creations during interpretation. The vertical axis shows the passage of time, moving downward, with the marked times being discussed in the text. As each schema instance is activated by a goal request, a schema instance is created. Each vertical line in the figure indicates the duration of the activity of one schema instance. The label of the associated schema is above the line tracing its activation. The right-pointing arrows indicate when an instance requested a goal, causing the activation of another schema. The left-pointing arrows indicate when an instance responds to a goal. (The dots indicate which crossing lines are connected.) The horizontal placement of the trace of each schema's action is arbitrary; in other words, the length of these arrows (left and right) has no significance. Additional important events are indicated by circles on the schema instance trace line and by the associated label indicating when the event occurred (e.g., at time t_{15} the house-geometry hypothesis is added to the house interpretation and to STM). The list on the right indicates the entities that are added to STM as a result of the schema activity.

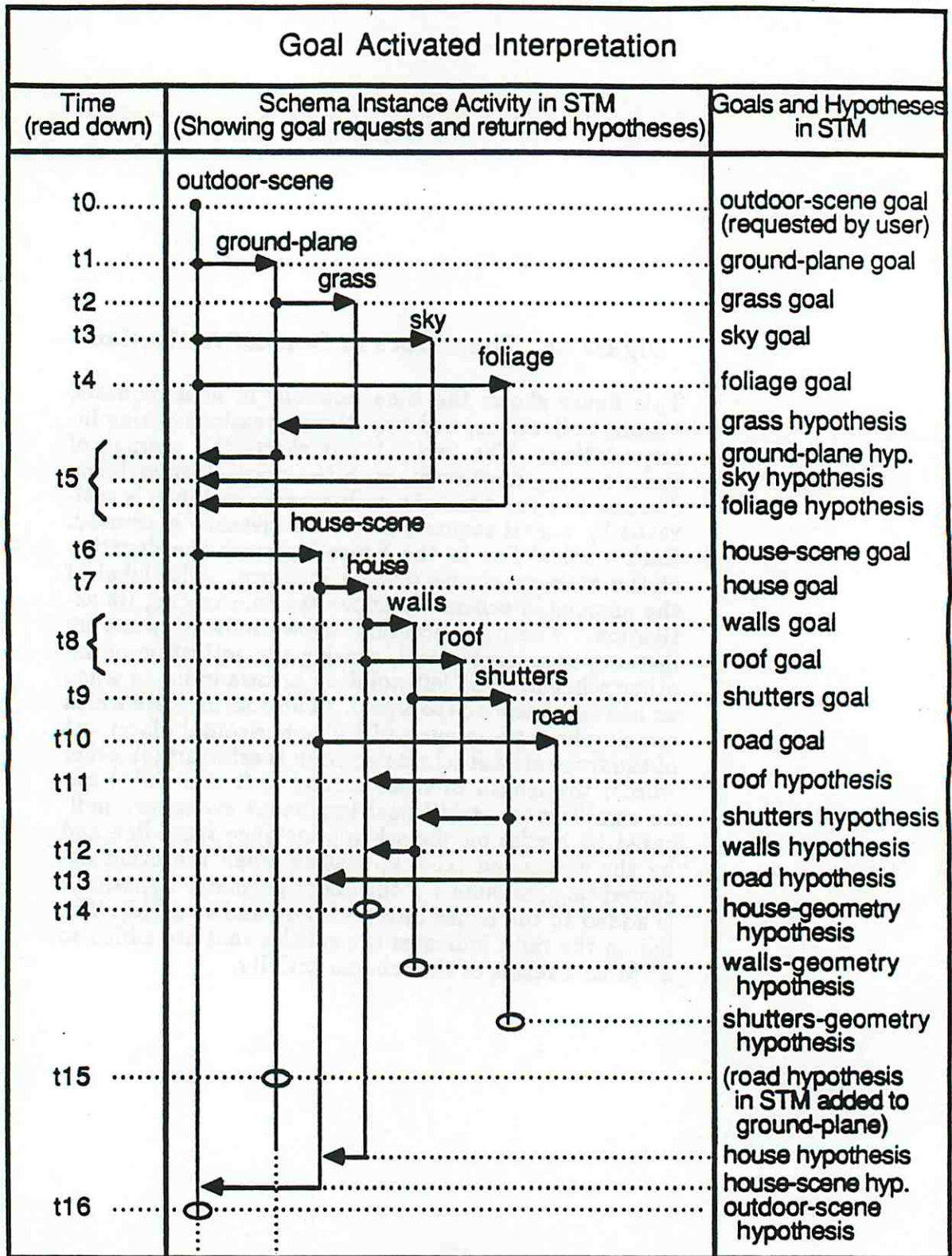


Figure 45.

respective interpretation strategies (see the points labeled t_1 , t_3 , and t_4 in Figure 45). Figure 46 shows the relations among the active schemas in LTM and the instances in STM corresponding to the interpretation after these three goal requests have been issued.

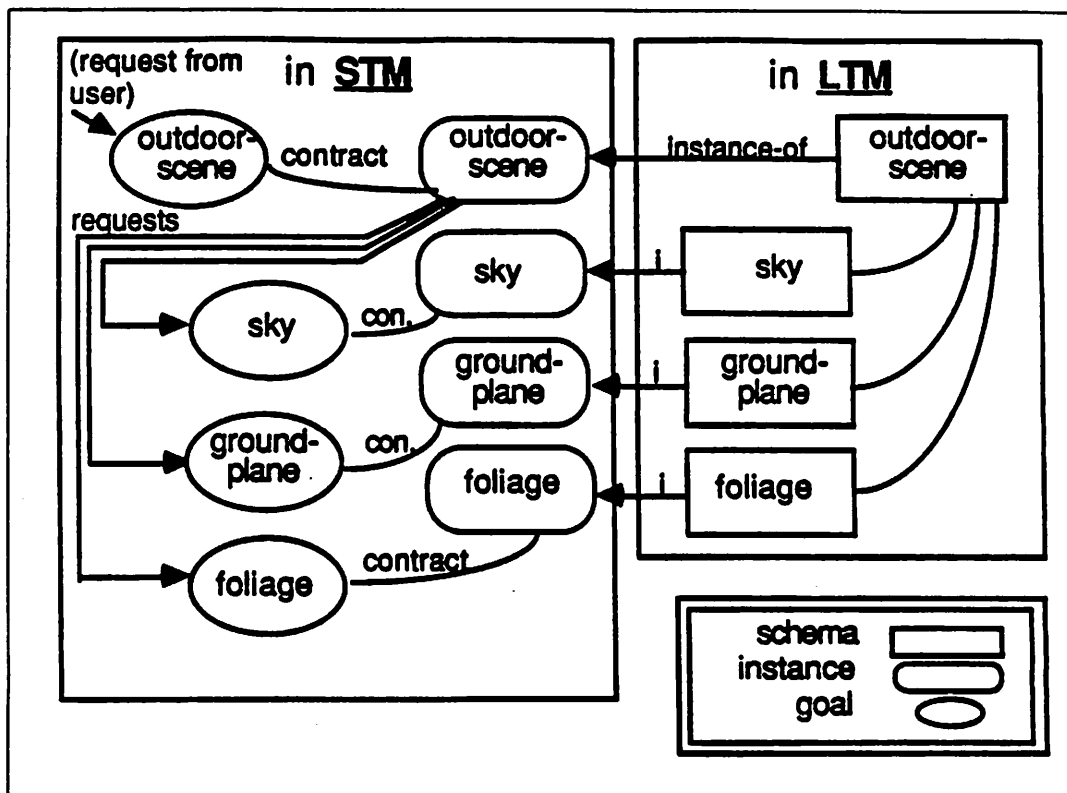


Figure 46. Relation between LTM and STM Nodes

As the interpretation proceeds, STM contains nodes for the active schema instances. This simple illustration shows these relations early in the interpretation. The labels on some of the arcs are abbreviated: *i* for instance-of and *con.* for contract. Note that the schema instances requesting a goal are linked to the goal nodes (the request line), and the goal nodes are linked to the schema instances working on the goal (the contract link); thus, the requesting instance and the goal-satisfying instance are linked through the goal nodes; these links are used for communication.

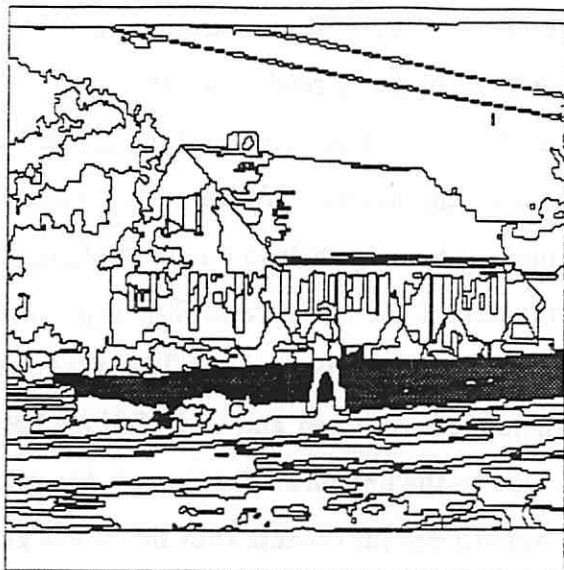
The point t_2 in the time trace illustrates top-down activation. The outdoor-scene interpretation strategy requests the activation of the ground-plane schema and the ground-plane schema requests a grass goal, causing the activation of a grass schema (see Figure 46).*

We also see (at t_4) that several instances can be simultaneously active. At this point in the interpretation there are five active instances: two are suspended, the outdoor-scene and ground-plane instances, because they are waiting for a response to the goal requests; the other three (the sky, grass, and foliage schemas) continue to execute their interpretation strategies.

The interpretation strategies for grass, sky, and foliage make no additional goal requests and interpret the image directly, via their separate strategies. When completed, they each return a hypothesis for the object which is associated with labeled regions in the image. These resulting responses could have occurred in any order. (See the point t_5 in Figure 45; also, Figure 47 for grass, sky and foliage results.)

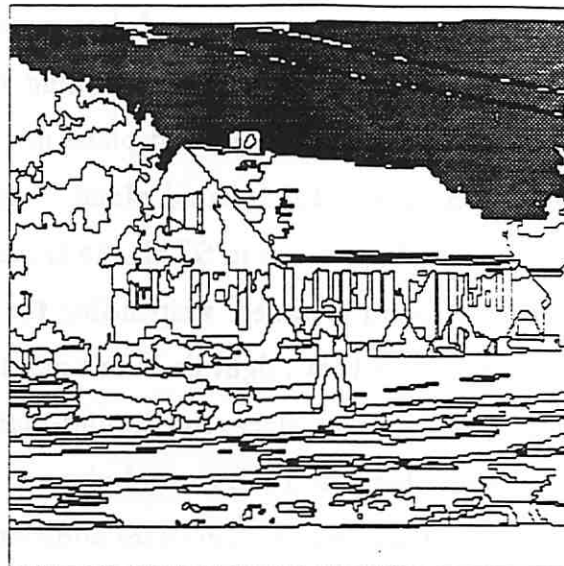
After the response of the grass instance, the ground-plane instance creates a ground-plane hypothesis incorporating the grass hypothesis. Since grass is a subclass of ground-plane, the image has two simultaneously valid labels for the associated regions. Eventually, the ground-plane hypothesis is returned to the outdoor-scene instance and included in the outdoor-scene hypothesis. In a similar way, each goal in the top-down cascade of requests is satisfied.

* This point in the time trace (t_2) also illustrates the asynchronous effect of the simulation. Note that in this particular run the grass schema was activated before the program steps in the outdoor-scene interpretation strategy which activate the sky and foliage schemas. There is no particular reason for this; it merely reflects the random selection in the simulation.



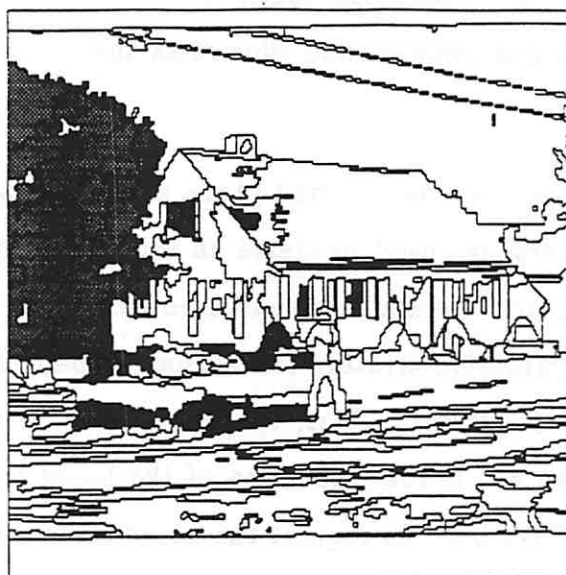
SEG301+ 308--EXTEND-GRASS-FROM-RULE (1-MAR-1985)

(a)



SEG301+ 310--EXTEND-SKY-FROM-RULE (12-MAR-1985)

(b)



SEG301+ 312--EXTEND-FOLIAGE-FROM-RULE (1-MAR-1985)

(c)

Figure 47. Results for Grass, Sky, and Foliage Interpretation
 Initial interpretation consists of labeled regions in the image. (a) Grass, (b) sky, and (c) foliage partial interpretations from their respective strategies. The details of these individual interpretation strategies are given in Chapter 3.

The instance returning the hypothesis is free to continue processing. That is, the interpretation strategy of that instance need not stop; it can continue constructing additional hypotheses or refining those that it has already constructed. Such is the case with the ground-plane instance. In addition to returning the ground-plane hypothesis, the ground-plane instance remains active. When it has created its initial hypothesis in STM, the ground-plane schema goes into a loop in which a wait command is issued, suspending the execution of the instance until a hypothesis for an object that might lie in or upon the ground plane is posted to STM. When such a posting takes place, the ground-plane instance is resumed and the relations between that object and the ground plane are added to the network. In this way, the ground-plane instance accumulates additional hypotheses for objects that lie on the ground plane and adds these new hypotheses to the ground-plane hypothesis in STM. This exemplifies how an instance can continue to contribute to the interpretation after responding to a goal request and illustrates the use of STM for communicating information.

When hypotheses for sky, ground-plane, and foliage are returned to the outdoor-scene instance they are used to create an initial description of outdoor-scene. A hypothesis for outdoor-scene is constructed from that description and posted in STM. The interpretation strategy for outdoor-scene next attempts to satisfy goals for the related sub-classes. Thus, the second step in the outdoor-scene instance (see t_6 in Figure 45) is the invocation of the house-scene schema. An alternate strategy would have been to request the house-scene goal at the same time as the ground-plane, sky, and foliage goals. This was not done, however, for two reasons: the cost of the interpretation strategies of those first three schemas was relatively low compared to the cost of the house-scene schema, and the success of those three schemas permitted the establishment of the outdoor scene schema.

The outdoor scene strategy shows one application of the "focus of attention" concept. Generally, in our scene domain the sky and ground plane hypotheses are readily verified and easily constructed. They provide a frame of reference for the outdoor scene hypothesis and account for a large portion of the image. Thus, the outdoor-scene strategy can concentrate on smaller portions of the image, use assumptions about where the ground-plane is, and request those schemas with strategies that are more context dependent and expensive.

The question naturally arises: "What if the sky and grass schema either returned contradictory results or failed to form hypotheses?" The general options that were suggested earlier are: retry the same interpretation strategy with different parameters, try a different strategy, or abandon the attempt. In particular, for this interpretation strategy, interpretation can not proceed (on an outdoor scene) without the sky hypothesis or the ground plane hypothesis. That is, we have defined an outdoor scene by its parts. This does not mean that interpretation would necessarily come to a halt. As illustrated in the next section, bottom-up activation could cause interpretation of much of the image.

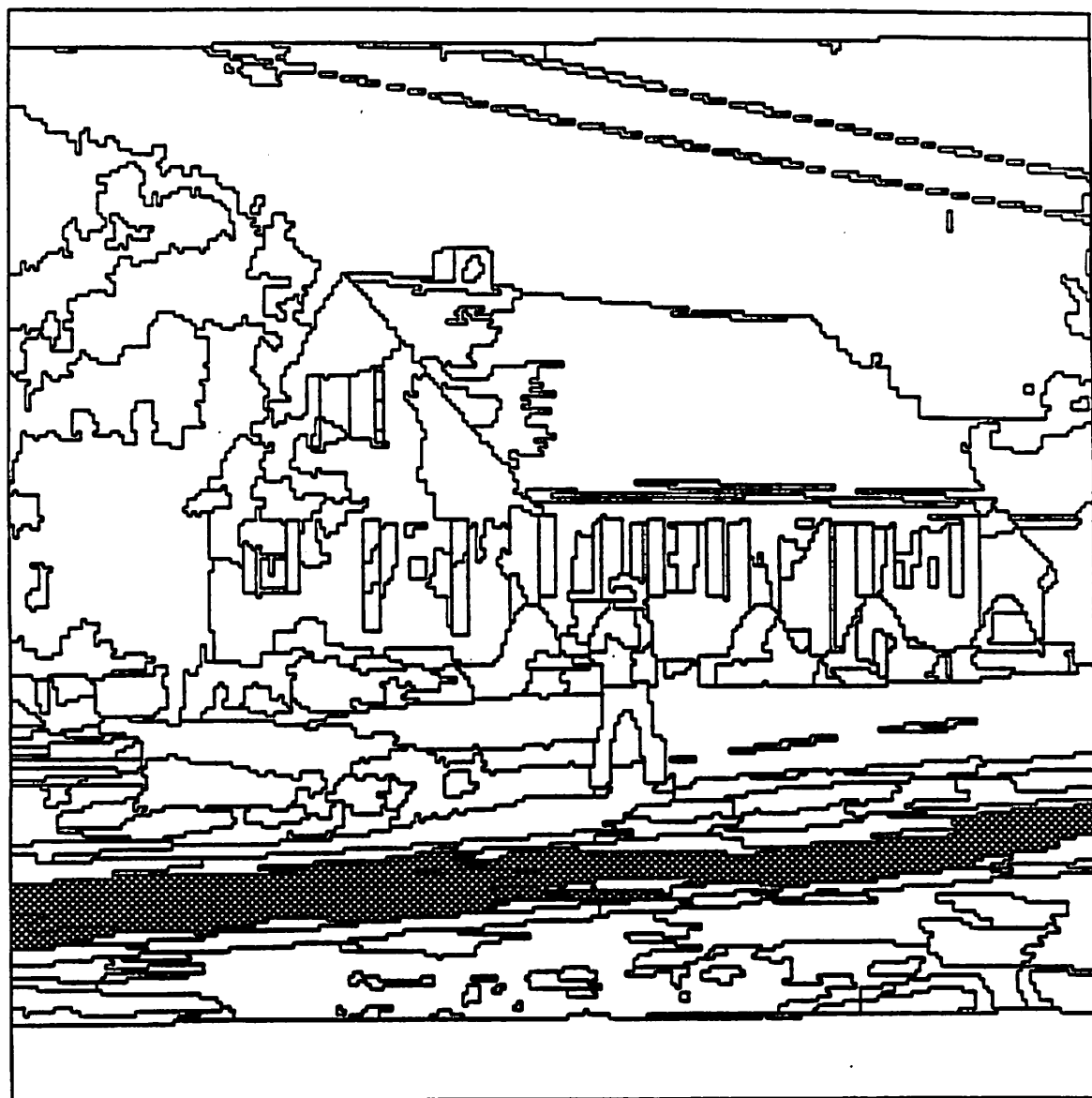
In the case where a richer network with schemas is available another strategy is to activate the schemas for each type of scene. If we know that the photograph is of an outdoor scene, then we might have schemas for several types of outdoor scenes. Some likely outdoor scenes, for example, are a house scene, a road scene, a forest scene, and a pasture scene. If there were a limited number of possible scenes in the domain and schemas for each of them, they could be activated in parallel and each allowed to proceed. The common features of those hypotheses that were successfully returned would then be characteristics of the outdoor scene. Conflicts would have to be resolved by the outdoor scene schema.

The house-scene schema instance is controlled by an interpretation strategy that first finds its parts, specifically, house and road. This is similar to the basic strategy employed by the outdoor-scene schema. To identify house and road, the house-scene interpretation strategy makes the goal requests that cause the creation of a house instance and a road instance (Figure 45, point t_7 and point t_{10}).

Again we see the branching out of parallel activation in the time line. The interpretation strategy associated with the road instance does not cause the activation of further schemas and eventually returns a road hypothesis (see Figure 48 and t_{13} in Figure 45). In this demonstration, the road instance is active during the time that the roof and shutters instances are active; during this time the instances for outdoor-scene, ground-plane, house-scene, and house remain suspended.

At this point in the interpretation we have seen the illustration of three major points of this example: 1) the overall interpretation proceeds from the actions of several interpretation strategies operating in separate schema instances, 2) independent interpretation strategies can be simultaneously active, and 3) dependent instances can communicate through goal requests.

In addition to communicating through goal requests and the responses to them, instances also communicate through STM. An example of this occurs in the ground-plane interpretation strategy. Road is one of the objects in the ground-plane. Recall that the ground-plane schema is suspended waiting for the creation of those hypotheses in STM that represent objects which could potentially lie on the ground plane. The creation of the road hypothesis is one of these events. The road hypothesis is posted to STM when it is completed by the road instance. In response to this event the interpretation strategy of the ground-plane instance adds the road hypothesis to the ground-plane hypothesis of the outdoor-scene (t_{15}).



SEG301= 579--DARK-ROAD-ROAD-EXTENSION (1-MAR-1985)

Figure 48. Road Interpretation Results

This figure shows the image regions labeled by the road interpretation. Details of the interpretation strategy for road are given in Chapter 3.

Up to this point in the example, we have examined relatively independent schemas. We now examine a collection of more closely interacting schemas. The creation of the house instance (t_7) starts the interpretation process for house. In this example, the interpretation strategy for house first selects the method based on finding the roof; thus, a request for roof activates the roof schema (approximately t_8), ultimately creating a roof hypothesis (see Figure 49 and t_{11} in Figure 45). In addition to, and parallel with, the activation of the roof schema, the house interpretation strategy requests a goal for walls (also at approximately t_8). The cascading of requests for goals and the corresponding instances echoes the part-whole relations in the schema network in LTM (Figure 50).

As discussed in Chapter 3, the house, walls, and shutters schema instances interact through the use of goals and hypotheses in STM. One specific type of hypothesis which is of interest is the hypothesis for the geometric structure, such as is constructed for the house and walls. Such hypotheses are generated to facilitate the interaction between their respective interpretation strategies. They are also incorporated into the object hypotheses; however, their separation in STM allows the interaction described in the following paragraphs.

In the schema network, house is part of house-scene, the walls are part of house, and shutters are part of the walls. Thus, each respective instance has issued a goal based on the expectations that specific objects might be present: house-scene for house, house for walls, and walls for shutters (t_9). The shutters interpretation strategy responds to the shutters goal by creating several shutter-pair hypotheses. These shutter hypotheses are returned to the walls instance which uses them to get the image area for the house walls (see Figure 51). The walls instance then waits for the house-geometry hypothesis, which is created by the house instance (t_{14} in Figure 45).

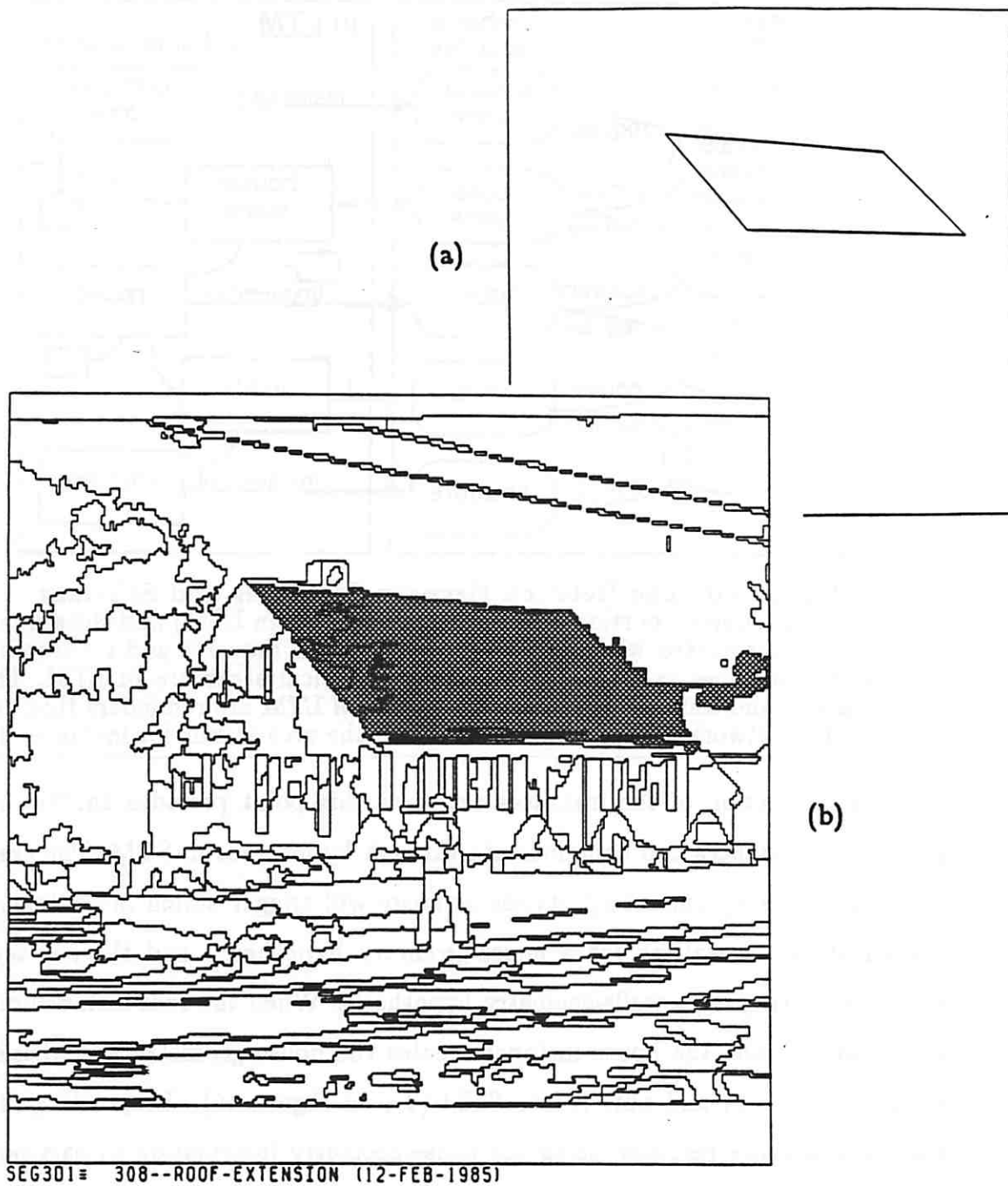


Figure 49. Roof Interpretation

(a) A projection of the rectangle from the roof interpretation (part of the roof geometry hypothesis); (b) the regions labeled as part of the roof interpretation network.

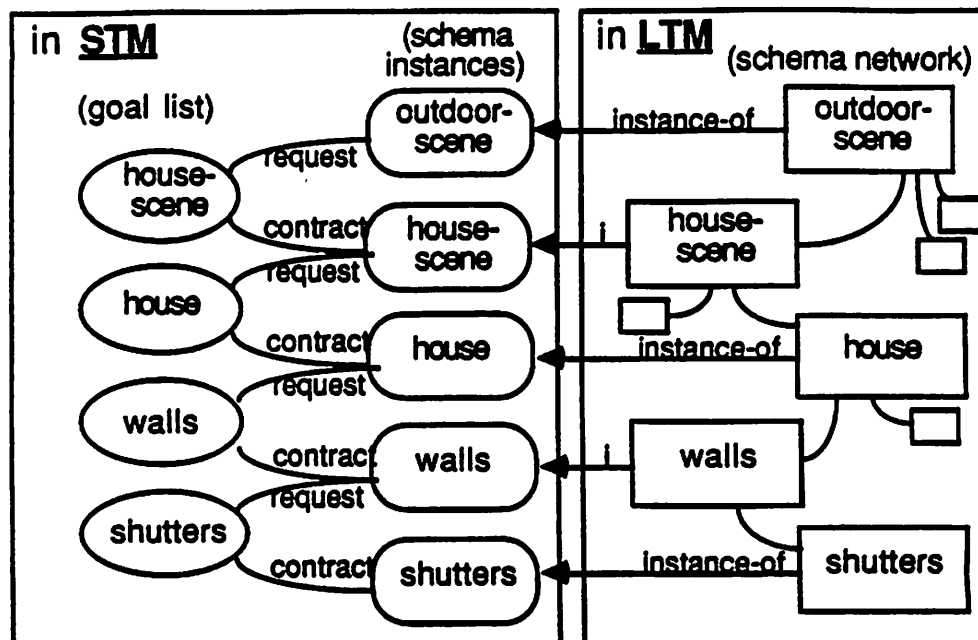


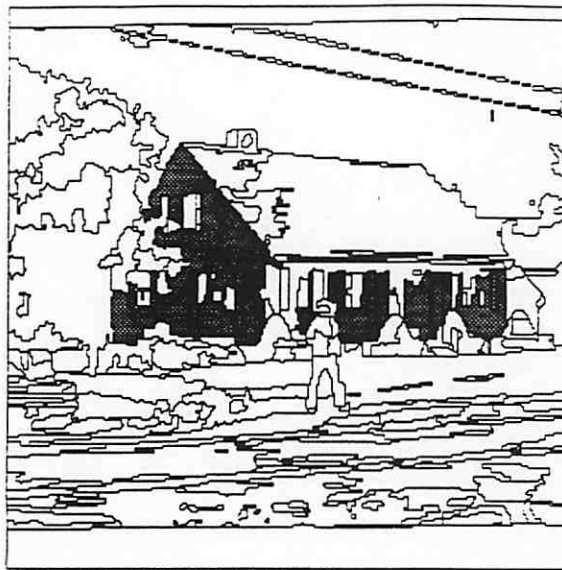
Figure 50. The Relation Between Instances and Schemas

This figure shows a portion of the schema network (in LTM) and the schema instances associated with it. The cascade of goal requests and contracting schema instances in STM reflects the hierarchical structure in LTM. The extra arcs and unlabeled boxes in the sketch of LTM are reminders that not all of the network is shown. (The label *i* on the arcs stands for instance-of).

The interaction of the instances active at this point provides another example of how instances can communicate through hypotheses in STM. The creation of a hypothesis by one interpretation strategy will trigger action in another. The walls instance is waiting for a house-geometry hypothesis, and the shutters instance is waiting for a walls-geometry hypothesis. When the roof instance returns a roof hypothesis, the house instance creates the house-geometry hypothesis (see Figure 52a and b) and puts it into STM (t_{14} in Figure 45). Responding to this, the walls instance resumes, using the house-geometry information to partition the house wall regions, and creates the walls-geometry hypothesis, which is also posted in STM. The posting of the walls-geometry hypothesis starts the shutters instance and the three-dimensional spatial information in that hypothesis used to separate

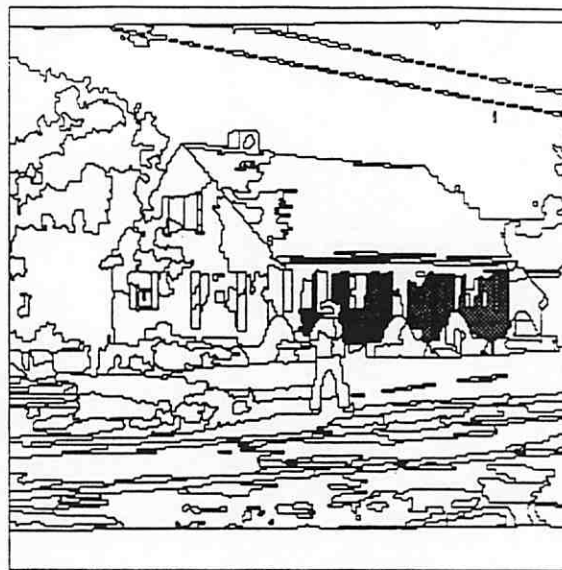
Figure 51. House Walls

(a) The image area labeled by walls hypothesis; (b) the regions associated with the front wall hypothesis; (c) the labeling for side wall. The separation of the labeled regions into two groups (for the two walls) is guided by the information about house geometry which is part of the house hypothesis.



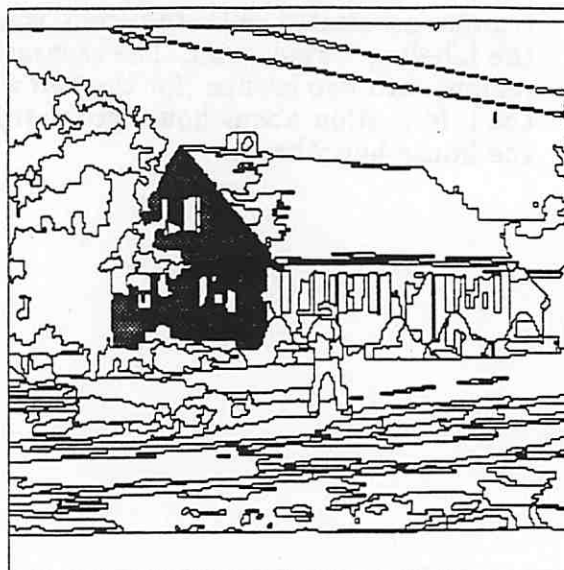
SEG301: 584--HOUSE-WALLS | 1-MAR-1985|

(a)



SEG301: 586--FRONT-WALL-IMAGE-AREA | 1-MAR-1985|

(b)



SEG301: 587--LEFT-SIDE-WALL-IMAGE-AREA | 1-MAR-1985|

(c)

Figure 51.

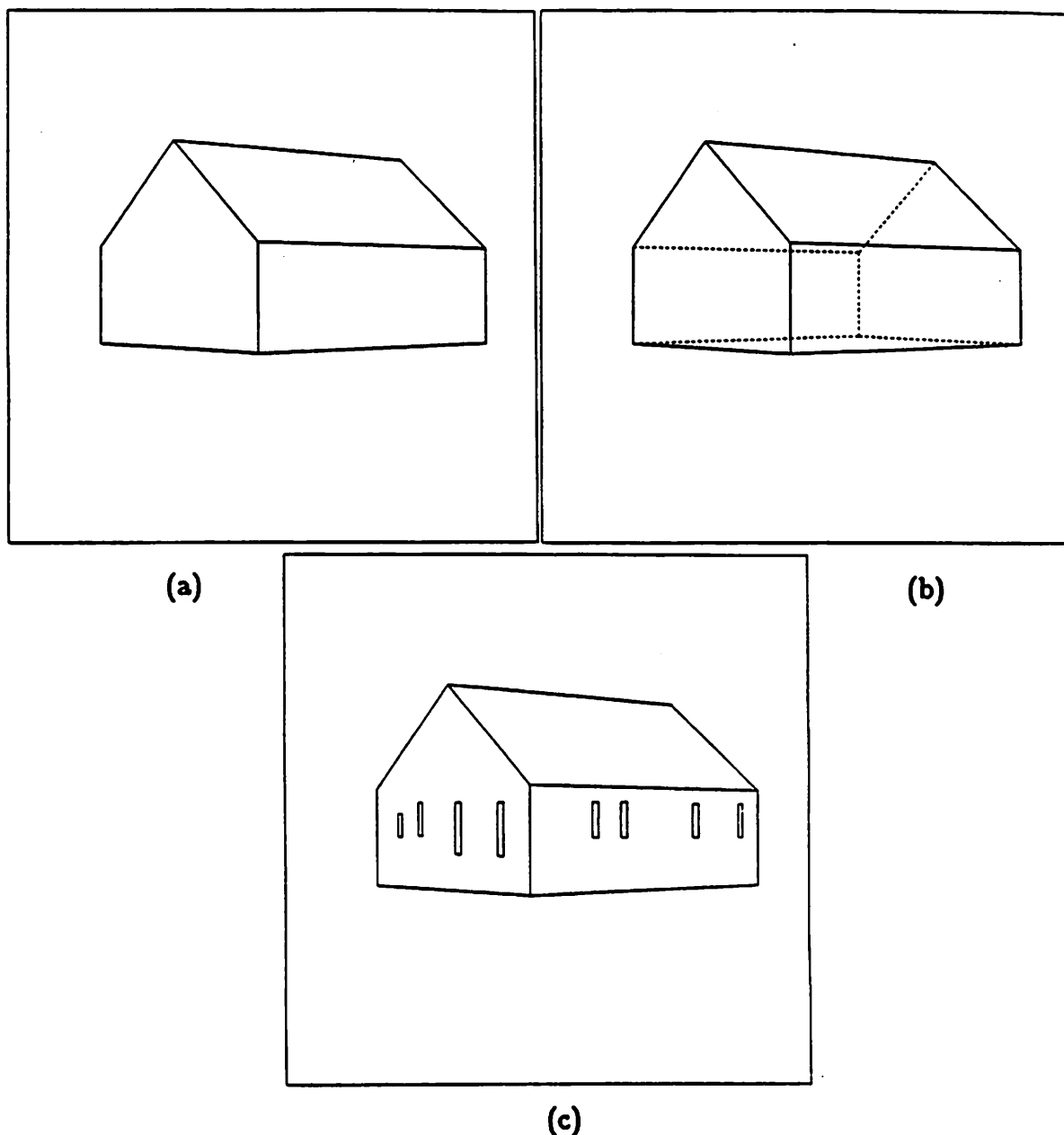


Figure 52. House Geometry

(a) A projection of the image-specific geometric model of the house (using the camera model and standard computer graphics techniques). (b) The same projection with the hidden lines removed. The partitioning of the walls shown in Figure 51 was derived from this geometric information. Since each wall is separately represented, the geometric information can be used to separate shutter-pairs and assign them to the appropriate wall in the geometric instance. (c) Illustration of the spatial effect of associating each shutter pair with the appropriate wall. See Figure 53b for the interpretation network associated with the house.

the shutter pairs and associate them with their respective walls, as described in the interpretation strategy for house walls, Chapter 3 (see Figure 52c). Thus, more complex interactions are achieved through interaction between hypotheses in STM and instances awaiting them.

The results of this interpretation of an outdoor scene is the creation of an outdoor-scene hypothesis. With the creation of its hypothesis the house interpretation strategy terminates, returning the house hypothesis to the house-scene instance. The house-scene interpretation strategy constructs a hypothesis and returns it to the outdoor-scene instance. The interpretation strategy for outdoor scene adds this new hypothesis to the outdoor-scene hypothesis and the interpretation is complete (t_{16} in Figure 45). Note how the process of hypothesis construction echoes the tree of goal requests and the descriptive structure in LTM.

The simulation of schema activity terminates when there are no further actions to be taken; in other words, when all the active interpretation strategies either have no additional actions or are waiting for further hypotheses. In this case, since the system state can not change, we terminate the simulation. Note that not all the interpretation strategies will have terminated. An interesting metaphor for this type of system is that of a real-time operating system. Some tasks are essentially monitoring tasks: they are activated when new events occur (in this case, hypotheses in STM). Other tasks are serial programs: they do their jobs, halt, and are removed from the system. In this example, the latter part of the ground-plane interpretation strategy is a monitoring task. When the simulation halts, it is still waiting for related hypotheses. An interesting interpretation system, and one that will be the topic of future research, is one in which there are many more schemas with interpretation strategies like the one illustrated by the ground-plane schema. In such a system, each of these "monitoring" schemas would be

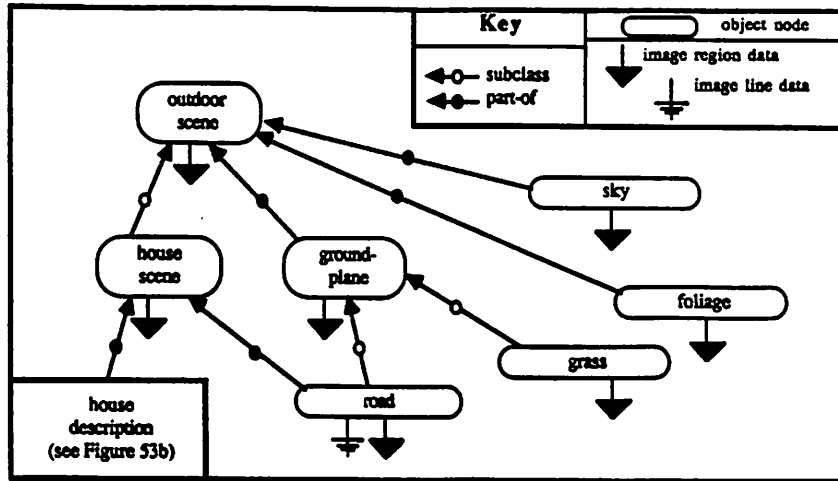
"tracking" the relevant changes in STM and continually updating its local part of the interpretation.

The resulting outdoor-scene hypothesis is a network describing the image (see Figure 53). Each object in the image is represented by a node in the network and those nodes are the object hypotheses that were collected in response to the goals. The principal relations in the network are the part-whole and class-to-subclass relations that were initially used to guide the interpretation. Thus, for example, the outdoor-scene hypothesis contains the hypotheses for sky, ground-plane, and foliage via part relations. Further, it contains the hypothesis for house-scene via a sub-class relation. The house-scene hypothesis, in turn, contains the hypotheses for house and road. In addition, the interpretation network provides information on other relations; for example, there is information about which objects are on the ground plane and the relations of each shutter to its respective shutter pair. Each hypothesis developed in the interpretation is placed in the interpretation network by its relational links to other hypotheses.

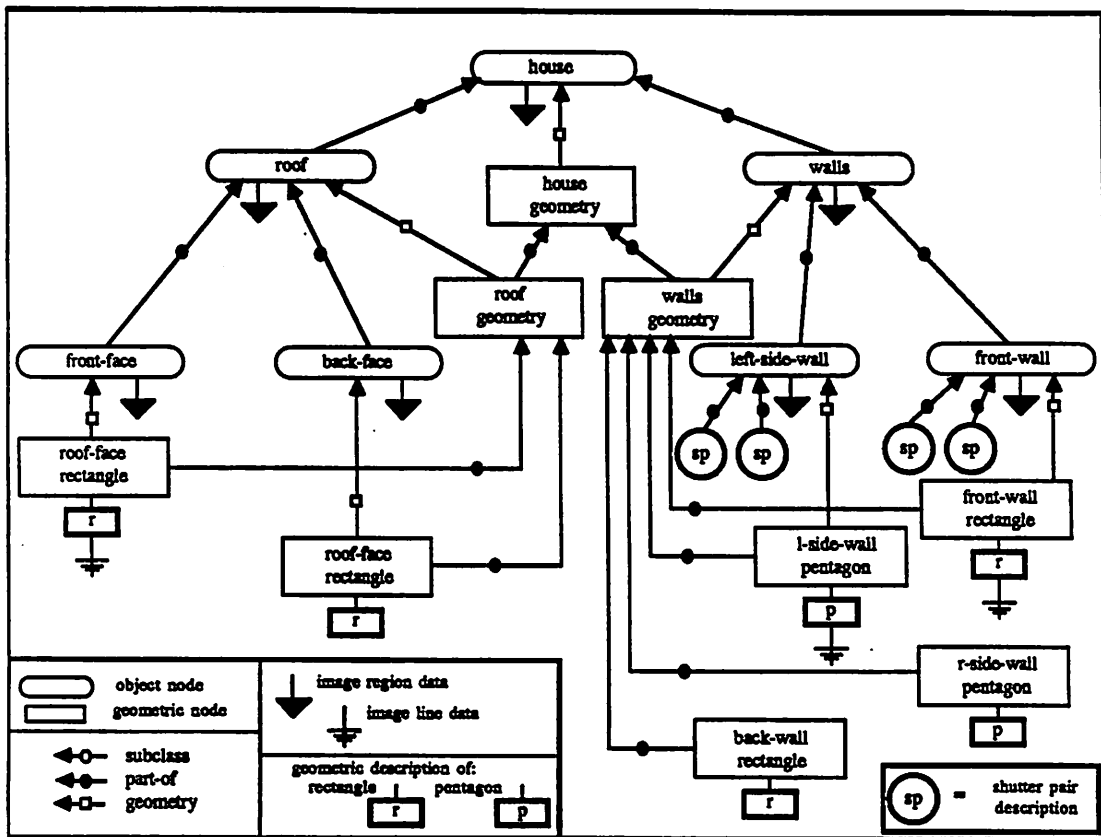
We can construct an image of the scene corresponding to the original image by using the spatial and geometric information in the interpretation network. Such a synthetic image can be used to evaluate the extent and quality of the interpretation. Figure 54 shows the "interpretation image" for this interpretation. It was obtained by combining the geometric information in the house, walls, roof, and shutter hypotheses with image region information from the hypotheses for the other objects in the scene. Note that there are additional hypotheses that could be extracted via the interpretation strategies developed in Chapter 3 (e.g., the wire schema); however, these were not included in this experiment.

Figure 53. Final Interpretation Network

(a) The overall interpretation, produced by the interaction of the interpretation strategies of the active schema instances, reflects the original schema network in LTM. Compare with Figure 42. (b) The portion of the interpretation network associated with the house hypothesis. The shutter-pair descriptions were not expanded (for clarity). However, each shutter pair node points to two shutters and each shutter pair has an associated rectangular surface description. The rectangle and pentagon description are ordered sets of lines with three-dimensional endpoints. Note that each shutter pair is associated with a particular wall. (The labeling of the walls as "front," "side," etc. is arbitrary.)



(a)

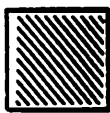
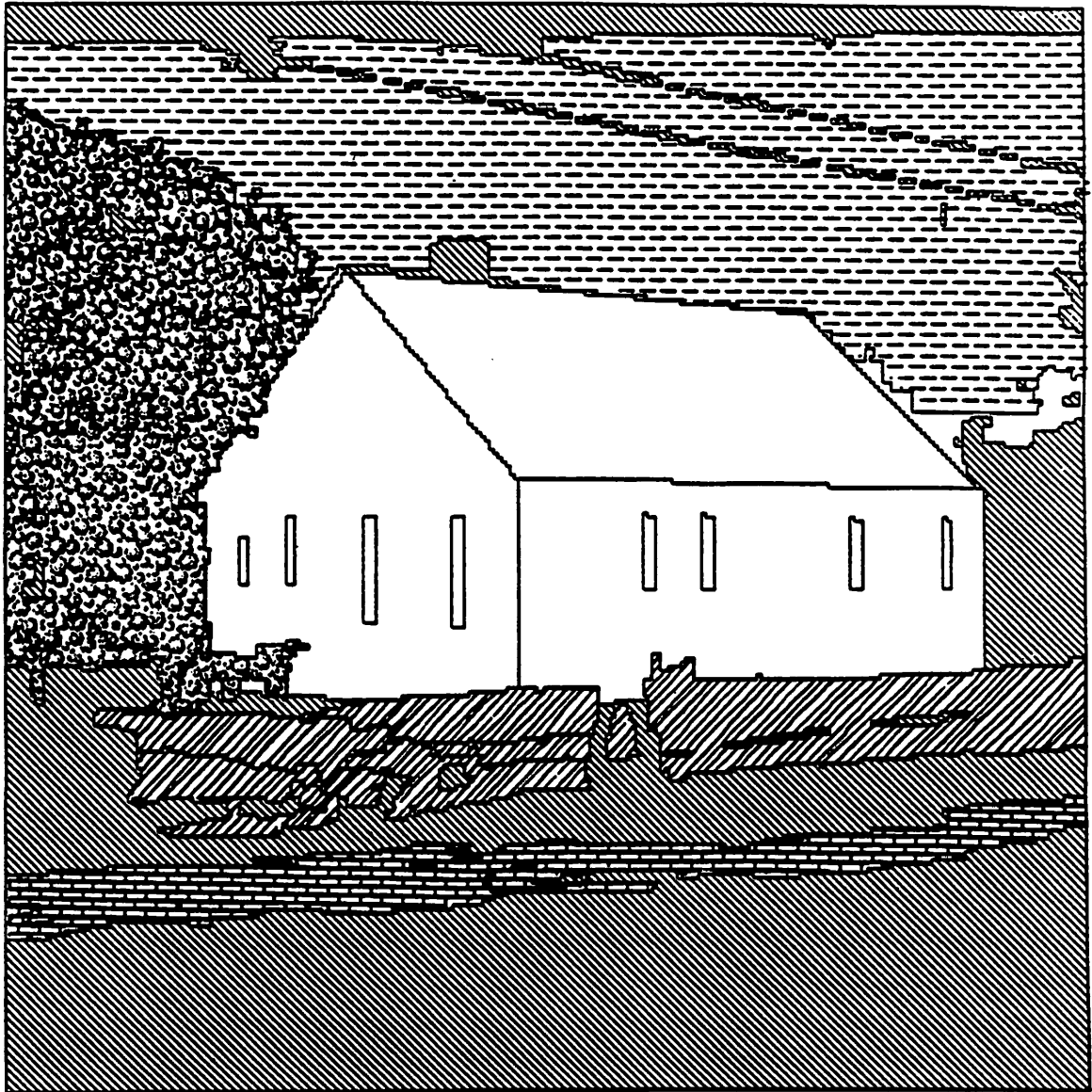


(b)

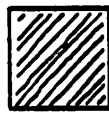
Figure 53.

Figure 54. Projection of Final Interpretation

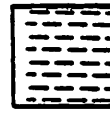
By using a few simple heuristics, the interpretation network can be projected back into the image plane. The projection of the geometry of the house, combined with the labeling for other objects from the network, produces an image of the interpretation. See Figure 56 for the detailed labeling of the house.



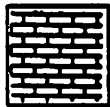
unlabeled



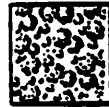
grass



sky



road



foliage



house area

(see Figure 56 for house labels)

Figure 54.

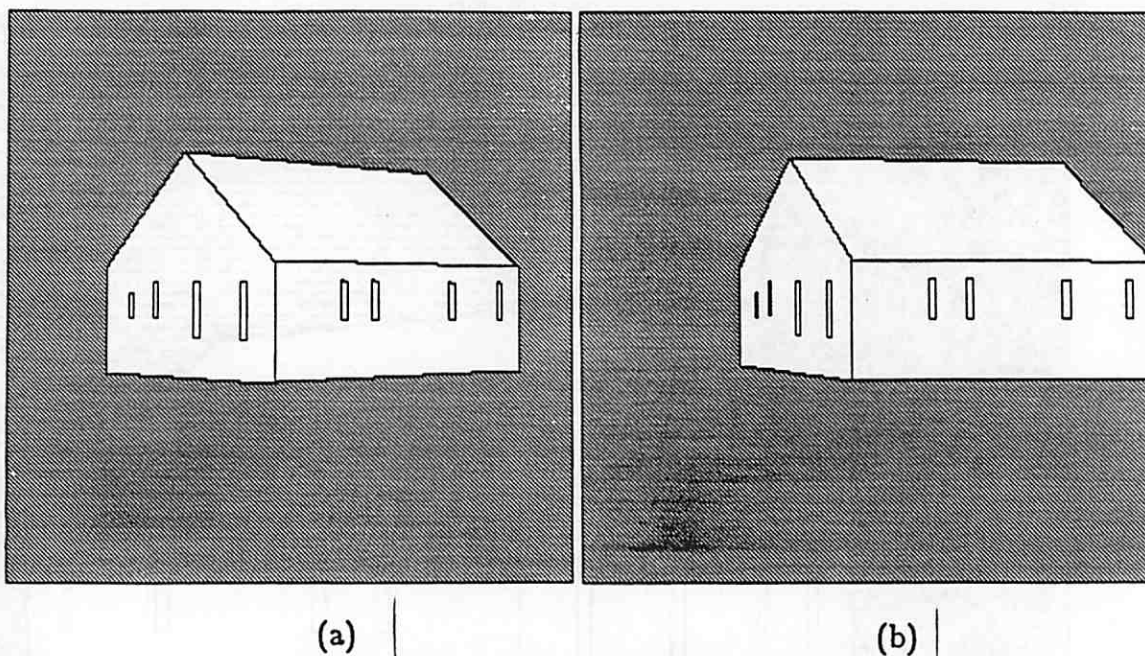


Figure 55. Projection of House Geometry

The scene-specific geometric description of the house is three-dimensional; therefore, it can be projected into the image plane in both the original position and in any new position. (a) The original object position and orientation from the interpretation; the normal to the front wall is $(0.44 \ 0 \ -0.89)$ in a camera-centered coordinate system. (b) The image-specific object model rotated about the corner edge closest to the camera, parallel to the y-axis (upright), towards the camera by 15 degrees. Observe that these projections are labeled pseudo-images: each image pixel has a label. The cross-hatched region is unlabeled and the image region within the house has labels - roof, front-wall, end-wall, or shutter - corresponding to the house part in the image.

Since the house hypothesis contains information about its three-dimensional structure, an image of the interpretation-specific model can be generated by projecting this three-dimensional description (see Figure 55a). It is also interesting to note that projections of other viewpoints can be generated (see Figure 55b) using standard computer graphics techniques.

Objects which do not have three-dimensional structural information as part of their interpretation can still be included in the interpretation image. The interpretation of every object includes a set of those regions from the original segmentation which correspond to the image area associated with the object. Thus, the location of the object in the interpretation image is represented by that set of regions. Where two objects have sets of the label regions that overlap, the outdoor-scene schema resolves the conflict using heuristics derived from the assumptions about the camera position described in Chapter 3.

The projection of the house is combined with the region labelings of the other hypotheses by using a distance estimate associated with each object. The distance estimates are used primarily to decide which regions should be projected as being "in front of" the projections of objects with geometric information (the house) and which regions should be "in back of" such objects. The distance estimates are obtained from a simplified camera model which we introduced in Chapter 3. In addition, where possible, objects known to be on the ground plane are associated with the projection of the ground plane (a horizontal plane in space) and objects known to be in the "background" are assigned an infinite distance. Finally, the sky is assumed to be above the camera and all other objects in the scene, and an infinite plane parallel to the ground plane; these assumptions have the effect of making the sky a background object.

There is a problem with how to treat the projection of objects on the ground plane. While some objects are known to lie in the plane of the ground plane, other objects might well be standing upon the ground plane. In this projection we made the simplifying assumption that each object could be sufficiently represented by using a simple z-buffer algorithm for producing the projection. In this algorithm, all the pixels in the regions in question are assigned a depth and only included in the final image if they are closer to the viewer than any other pixel. Since our

estimate of distance for each object area (merged region) is only a single number for the whole region, this has the effect of making the projected regions look like "cut-outs" when they are projected to the image. This is most noticeable in the relation between grass and house; that is, the relation "the grass is-in-front-of the house" gets explicitly represented in the projected image. A more natural image, and one that certainly can be produced from the information in the interpretation network, could have been produced if these objects were projected as lying on the ground plane.

In Figure 54 the combined projection of the labeled regions and geometric instance of the house demonstrate the overall effect of the interpretation. The areas of this synthetic image which are filled with a crosshatching are uninterpreted areas, and the labeled areas are those interpreted as parts of the outdoor scene. The details of the labeling of the house image are shown separately in Figure 56. If we discount the top and bottom border areas (which appear as spanning horizontal regions in the segmentation), then just over two-thirds of the picture area is interpreted. The identification of the major objects is represented in the interpretation network.

Several of the uninterpreted portions of the image are equally interesting. Some areas are unlabeled because the individual interpretation strategies failed to identify all the image portions belonging to the object. This is especially true of the "unstructured" objects such as grass and foliage. While the projection shown here accurately shows the image areas actually identified, it could be argued that the interpretation system should "hallucinate" the unidentified image portions of such objects in a manner analogous to the hypothesizing of the hidden structure of the house. For example, the grass on the ground plane is enough evidence for grass (in general) that the system could hypothesize that there is grass "everywhere" on the ground plane and that any unidentified image area which "let the ground plane

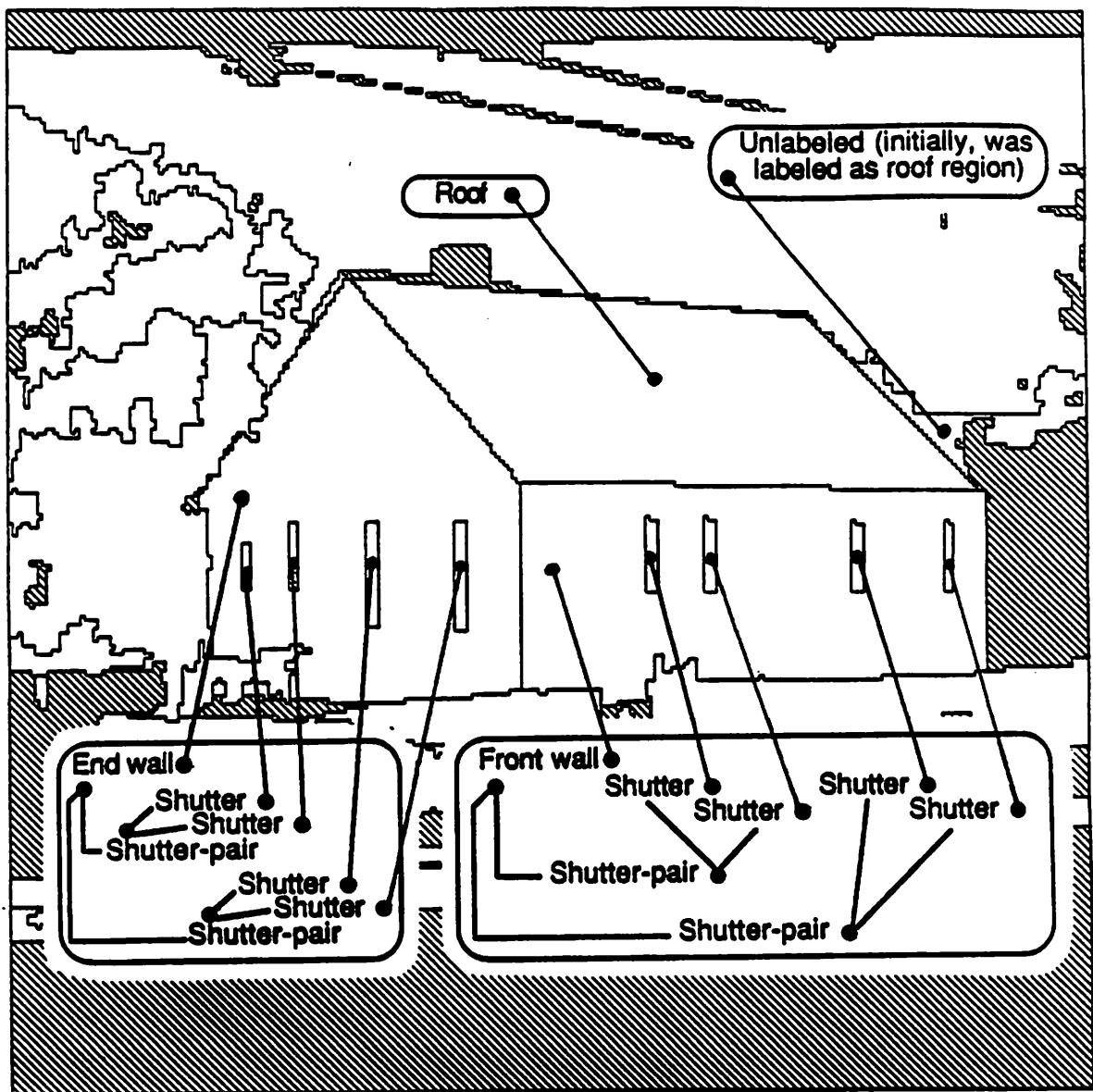


Figure 56. Projection of Final Interpretation of the House

This figure shows the details of the labeling of the house in the projection of the final interpretation. Note the structure of the shutter pairs and the relabeling of the roof region that was originally mislabeled because of a segmentation error.

show through" should be labeled grass. In the same way, the regions interior to areas labeled foliage could also be labeled foliage. Each of these extensions is based on knowledge about the structure of the object. The use of such assumptions would allow the generation of a more complete interpretation which would result in a more convincing projection.

The general concept of letting the system fill in the details of structure when there is little evidence would also have helped fill in the interpretation of the road and the features of the house. Because the system exhibits the ability to place an overall hypothesis for each of these objects, such hypotheses could be used as a base for extending the interpretation using contextual clues. Two examples of interpretation strategies based on context, for wire and telephone pole, were illustrated in the closing section of Chapter 3. Similar interpretation strategies would enable the system to hypothesize the curbs of the road, the chimney of the house, the windows (based on shutter pairs), and the placement of gutters on the eaves. With such additions the interpretation could be extended to another level of detail.

The uninterpreted areas corresponding to the parts of the person serve to remind us that there is a general problem with this approach. The system does not interpret objects about which it knows nothing. There is no strictly bottom-up interpretation. Clearly a schema for walking-person could have been developed. The legs, for example, are evident in the intermediate database as two converging pairs of parallel lines, and there is sufficient image evidence to verify the position of the upper body. An interpretation strategy for such a schema could be of the same context-driven type as discussed; especially if an additional schema for the sidewalk were possible. On the other hand, this gap in the interpretation does suggest that schemas of a more general nature are needed to "fill in" the connection to bottom-up interpretation. One can imagine several schemas that would attempt to account for portions of the image that could not be explained as parts of objects. For example,

some of these might be an unidentified region schema, a potential surface schema, an unidentified object on the ground plane schema, or an unknown geometric structure schema. If such schemas were possible, they could be used to account for those portions of the image that have been left uninterpreted.

3. Data-Activated Interpretation

Schemas can be activated in one of two ways, either directly by goal request or indirectly by the occurrence of a key event. In the previous section, we examined the case of goal-directed activation; in this section, we offer an example illustrating an interpretation initiated by data events. Using the same image as in the previous section, this example shows how interpretation can follow from the activation of the roof schema. As part of the schema for roof, a key event is the occurrence of a long horizontal line. The presence of such events in the image is enough evidence to activate the roof schema, on the supposition that there might, in fact, be a roof.

Of course, using evidence as tentative as a long horizontal line to activate the roof schema increases the likelihood that the schema will be uselessly activated when no roof is in the image. In such a situation, however, the subsequent failure of the schema instance to produce an object hypothesis would keep the propagation of effects from such activation to a minimum. As an added precaution against unnecessary activation, top-down and bottom-up activation could be combined. We could have made the key event for the roof schema be a much more conservative "long horizontal line below the sky." The presence of a sky hypothesis is required to detect such an event; in turn, this requires that the sky schema be activated, either as a goal request from the outdoor-scene schema instance or through a key-event type of activation. Given that a sky hypothesis were available and the set of long lines could be filtered using the relation "below sky," then the resulting hypotheses could be used to activate the roof schema. A roof schema activated in this way would be less likely to be activated for an image in which there is no roof. Such

a conservative type of key event may seem a reasonable choice in a more complex setting; however, we chose to use the simpler specification of the key event for roof schema because it served to illustrate the points of this section.

To initiate this interpretation, it was necessary to add a special pre-processing step to filter image lines which selected long horizontal lines via a rule-based hypothesis generation function.* The addition of these generated hypotheses to STM triggers the roof schema. Figure 57, showing the time trace of schema activation, illustrates how the activation of the roof schema leads to the interpretation of the image.

With event-initiated activation, it is useful to know which event actually caused the activation. Thus, when a schema is activated by a hypothesis, a pointer to that hypothesis is put in the local memory of the schema instance, under a variable with the same name as the hypothesis type. This allows the interpretation strategy of the schema to determine (if it needs to do so) which hypothesis (if any) was instrumental in initiating the local interpretation. For example, during the interpretation illustrated here, the creation of a roof hypothesis will cause the activation of the house-scene schema. When this happens, the interpretation strategy for house scene will use the roof hypothesis supplied as the key-event hypothesis to "prime" the house schema when it makes a goal request for house. It is able to do this because a pointer to the roof hypothesis is supplied to the house-scene schema instance as the value of the variable "roof." Had the schema been otherwise activated, the variable roof would have no initial binding in the local environment of the schema instance.

* This preprocessing could have been part of a start-up schema; in fact, all of the preprocessing steps (including segmentation, long line extraction, and getting any goals from the user) could have been coded that way; however, this seemed to be stepping outside the realm of object-based schemas.

Figure 57. Time Trace of Data-Activated Interpretation

This trace shows the interpretation of the previous image where the initial event is the activation of the roof schema by image data rather than a user-supplied goal. The resulting interpretation is identical to that in the goal-directed example, but the order of schema activation differs from the ones shown in Figure 45. Note, also, the randomizing effect of the simulation in the activation of the sky, foliage, and ground-plane schemas: the sky schema completes before either of the other two are activated, unlike in the first example, where they were all activated before any of them completed. The activation order of the walls and road schemas differs for the same reason. Finally, note that the roof schema is not activated by the house schema because the relevant interpretation information is passed to the house schema, as explained in the text.

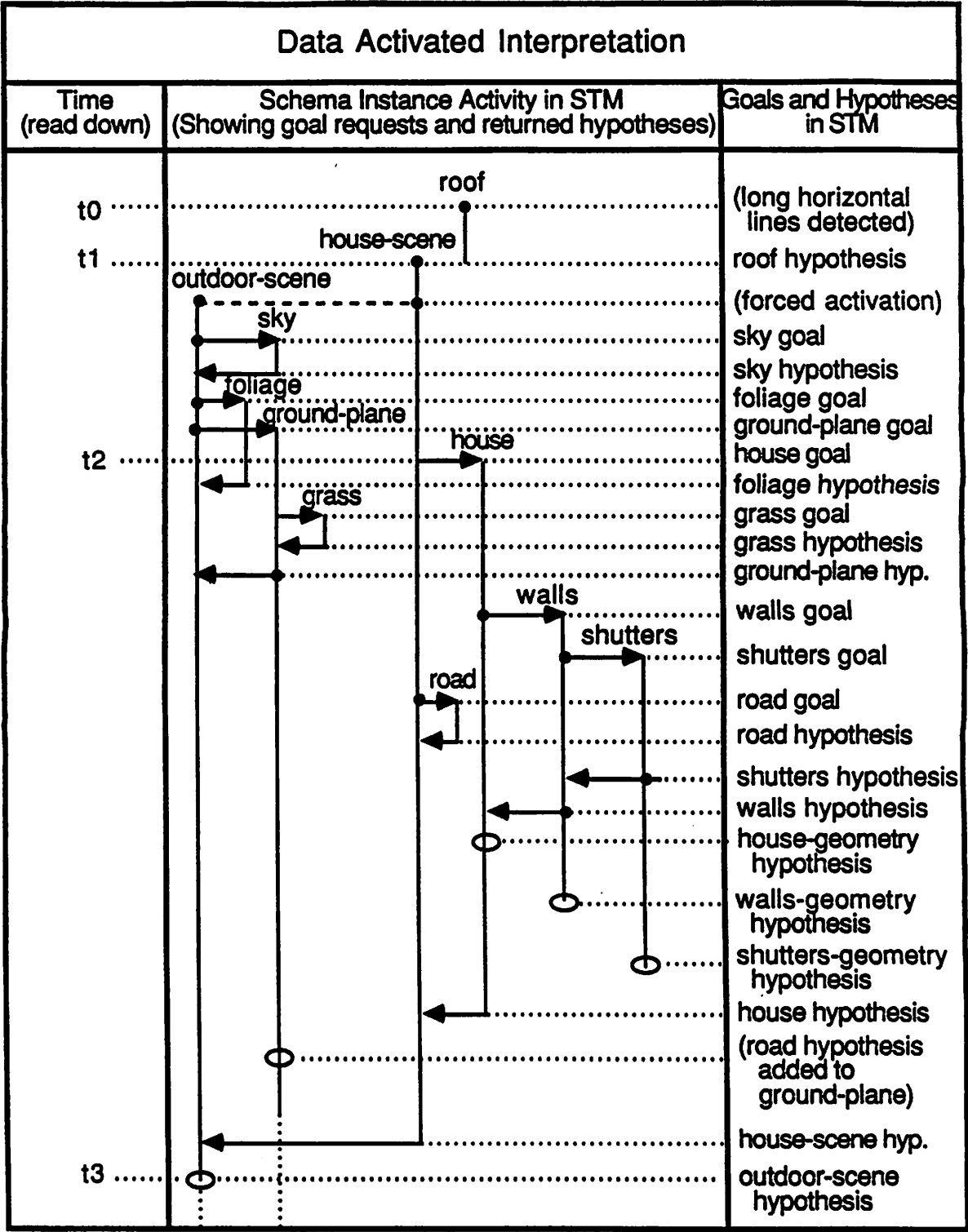


Figure 57.

Although this interpretation is initiated by the detection of a data event, much of the activity is goal directed. Once evidence for an object causes activation of the schema for that object, parts and sub-classes for that object are sought directly. Thus, once the house-scene schema is activated it proceeds to request goals and to integrate the resulting hypotheses exactly as described in the preceding section. This is also true of the other schemas shown in this trace of activation. Since a great deal of the activation of schemas in this example is exactly as in the previous section, we will mention only the differences, highlighting the effects of data-activation.

Activation by key events is used sparingly to prevent a combinatoric explosion of instances. In this example, only the roof and house-scene schemas have the ability to be activated by events in STM. When the roof instance creates a roof hypothesis and posts it in STM (see label t_1 in Figure 57), activation of the house-scene schema ensues because roof is a key event of that schema.

From the point of the activation of the house-scene schema (t_1) the interaction between schemas and the resulting interpretation is the same as in the goal-directed example, with two exceptions: the roof schema is not activated again and the outdoor-scene schema is activated from the house-scene schema. To communicate that the roof description already exists, the house-scene instance is initialized with a pointer to the roof hypothesis that is passed on to the house instance as a goal parameter. The house interpretation strategy recognizes that the roof hypothesis is already available and does not need to make a goal request for it. This process exemplifies the way in which instances communicate control information through goal parameters. The house-scene instance passed interpretation information through the goal request to the house schema instance (t_2).

We could have chosen to allow the event-directed activation of the outdoor-scene schema by the house-scene hypothesis. This would have worked in the same way as

the interaction between the roof and house schema instances, but it would have had the unfortunate effect of delaying the activation of the outdoor scene interpretation. Thus, because the house scene hypothesis is created so late in the overall interpretation, and because the outdoor-scene interpretation strategy performs many parts of the interpretation that are separate from the house-scene (and therefore can be performed in parallel), we chose to have the house-scene interpretation strategy directly activate the outdoor scene schema. The house-scene interpretation strategy requests activation of the outdoor-scene schema through a goal request. A house-scene is a type of outdoor scene; thus, existence of a house-scene instance implies that an outdoor-scene schema should be active. Otherwise, the active house-scene instance proceeds as described in the previous section and produces a house-scene hypothesis which becomes part of the outdoor-scene hypothesis (label t_3 in Figure 57).

4. Response of Schema Network to Scene Variation

The details of the interpretations discussed in this section show the response of the interpretation network to differences among selected scenes. In these illustrations, minor variations abound. There are differences in the part relations for the house. For example, some of the houses have shutters, some do not; in one image the roof is not visible. Further, each house scene is presented at different distances and in different views. Thus, differences in the images of each house and its surroundings are great. However, the general appearance of the scenes is similar (Figure 58); each of the photographs presents an image of a house scene consisting of a centrally located house with grass in front, surrounded by varying amounts of foliage, situated below a sky. Thus, each of the interpretations discussed in this section is qualitatively the same as the interpretation shown in Section 2 of this chapter; e.g., the general time sequence of schema activation and the interactions between schema instances is the same. Therefore, rather than concentrate on the details of

Figure 58. Additional Scenes

These images show the scenes used in the additional interpretation examples discussed in the text.

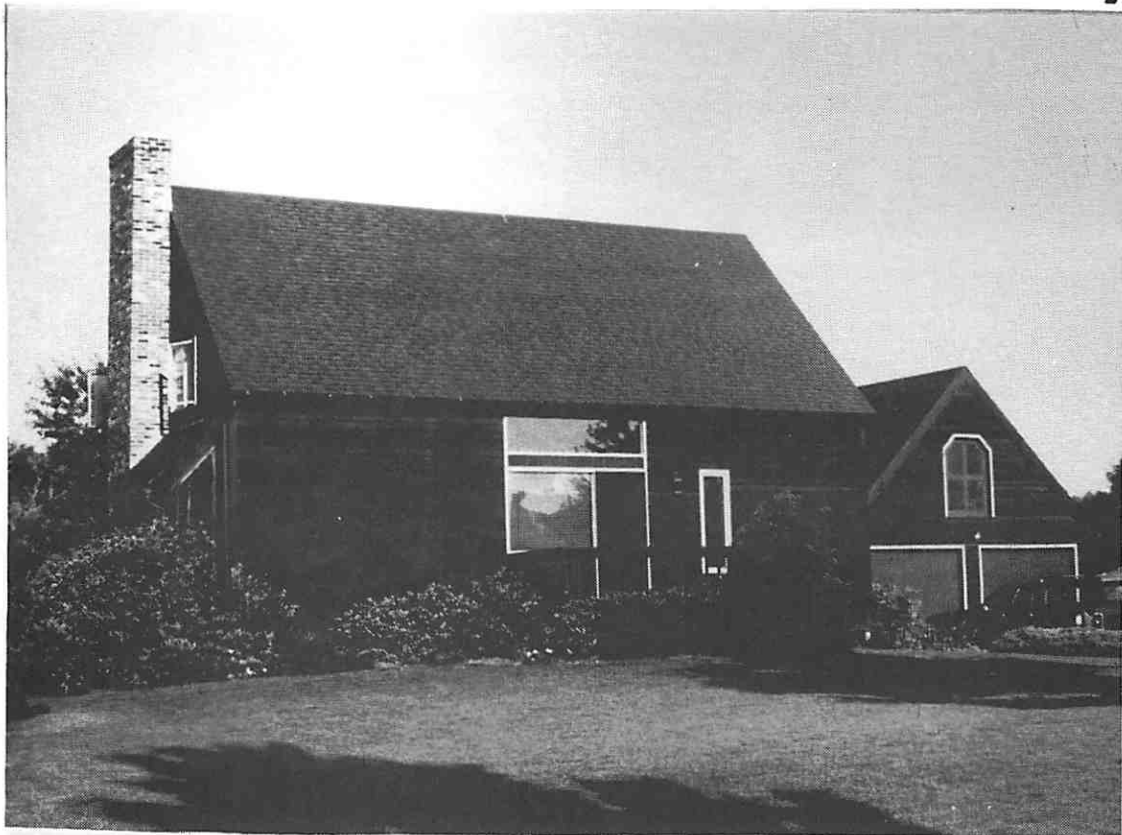


Figure 58.

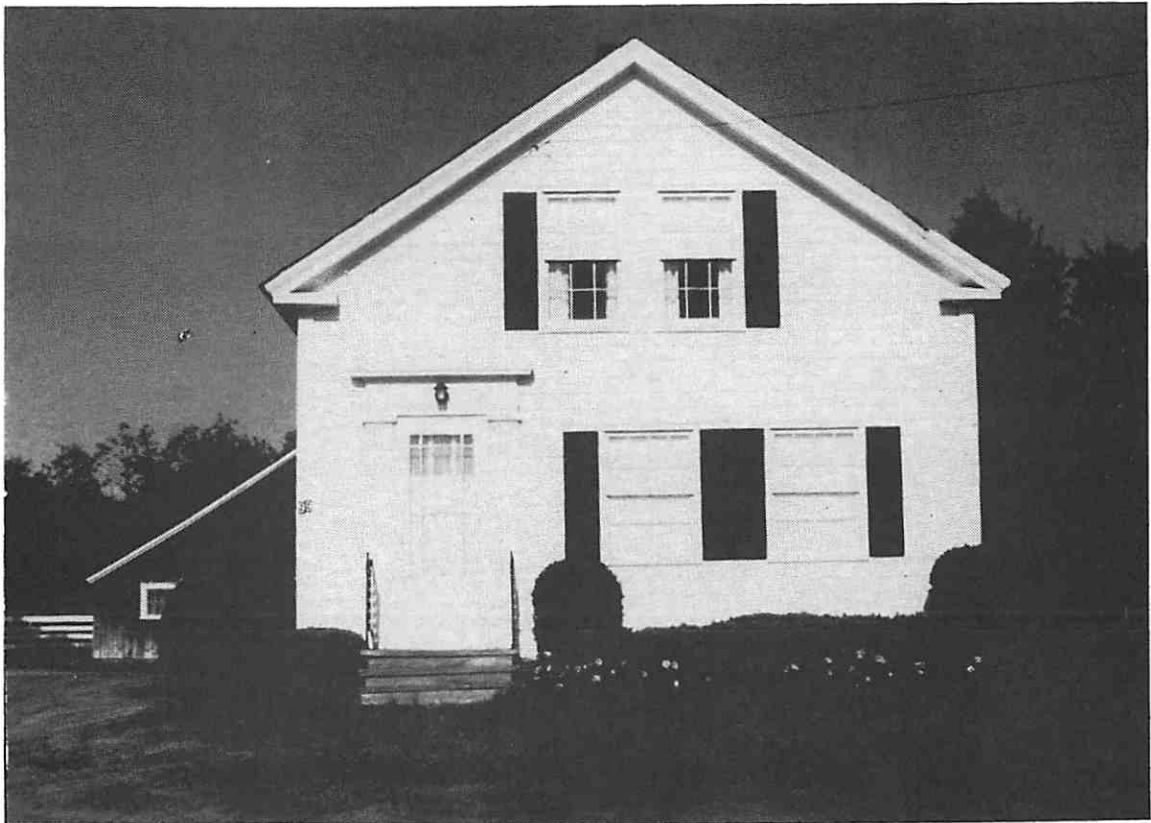


Figure 58 (continued).

the communications and interactions among the schemas, we will use these examples to examine two themes. The first one is that the current set of interpretation strategies has failings which are made obvious in these interpretations. However, our second theme is that the overall interpretations for these similar scenes, when they are obtainable, are similar over a wide range of viewpoints.

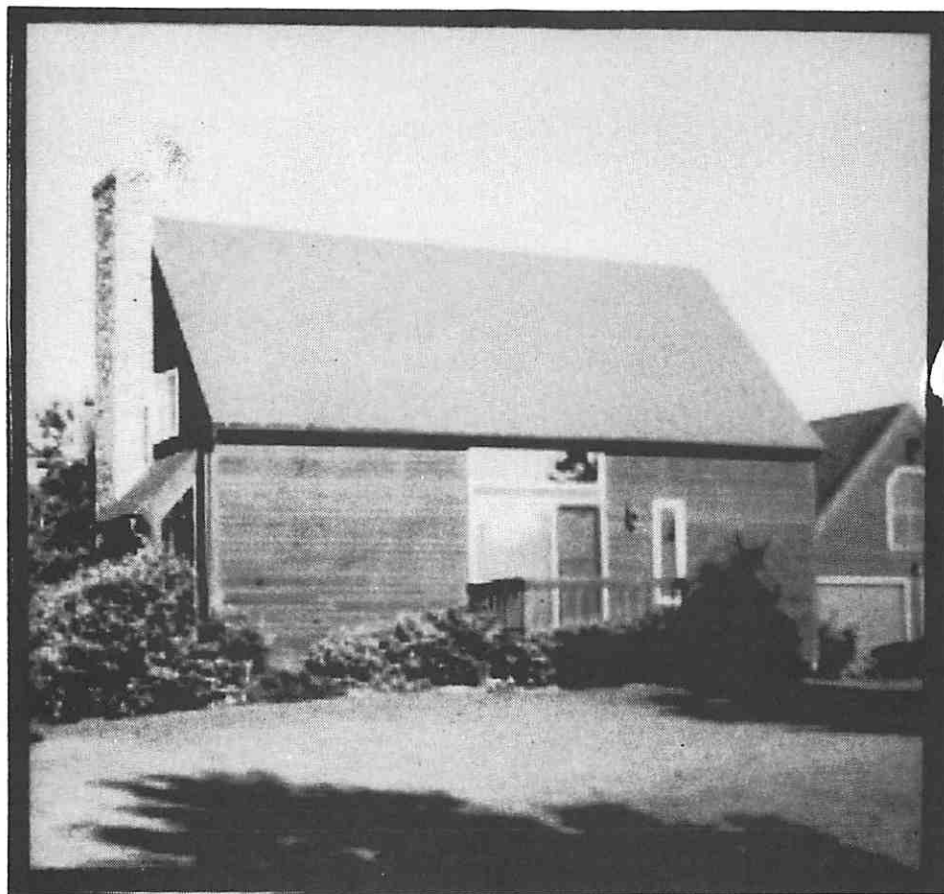
4.1 Viewpoint Variation

Let us first look at the case where a slight change in viewpoint produces variant images of the same scene (see Figure 59). At first glance, these images look almost identical. They each present a view of the side of a house sitting behind a short expanse of grass surrounded by shrubs with the sky clearly visible behind the house. This first impression, that the images are alike, is also reflected in the character of the interpretation process for the image. For each of these interpretations, the sequence of schema activation, the duration of schema instance activity, and the relative timing of goal and response among the schema instances are nearly identical, as is the structure of the resulting interpretation networks. The networks contain the same objects and the same compositional (part-whole) and scene specialization relations. The trace of schema activation and the resulting interpretation network are shown in Figure 60.

The variations among these images are manifested in the interpretation network as differences in the labeled images produced from the projections of each interpretation network (see Figure 61). These differences arise from the differences among the segmentations shown in Figure 59b. The image area labeled grass changes from image to image: for example, a shadow over the grass in one image alters the area covered by the grass label. In addition, the shifts in region location and divisions produced by the segmentation routines (due to the shift in position, variations in reflectance, and properties of the region segmentation algorithm) compounded by differences in the response of the interpretation strategies to these details, result

Figure 59. Similar Views of the Same Scene

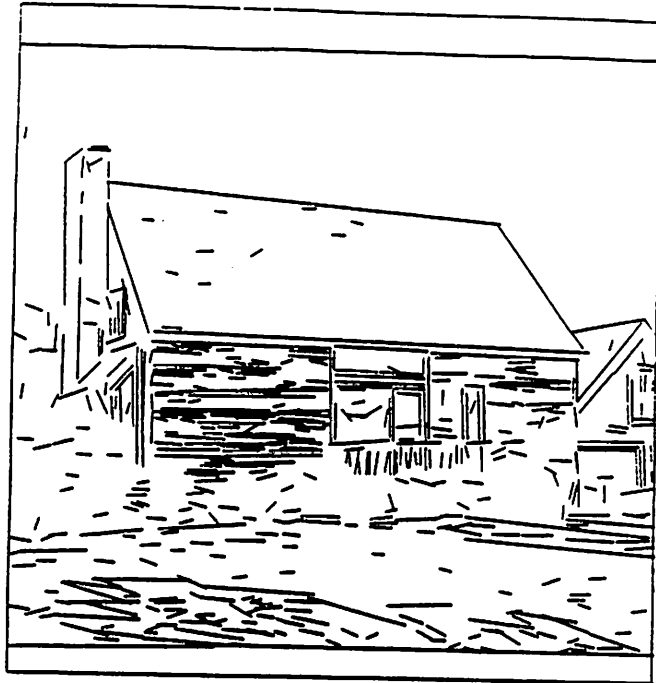
In this example the interpretation system constructs networks for three views of the same scene. (a) The digitization of the first image; (b) line data for the first image; (c) the segmentation for the first image. (d, e, f) Second image showing digitization, line data and segmentation. (g, h, i) Third image showing digitization, line data and segmentation. The differences in image details cause variations in the resulting labeled images; however, the interpretation networks exhibit the same general form as in the first example and are similar to each other.



(a)

Figure 59.

(b)



(c)

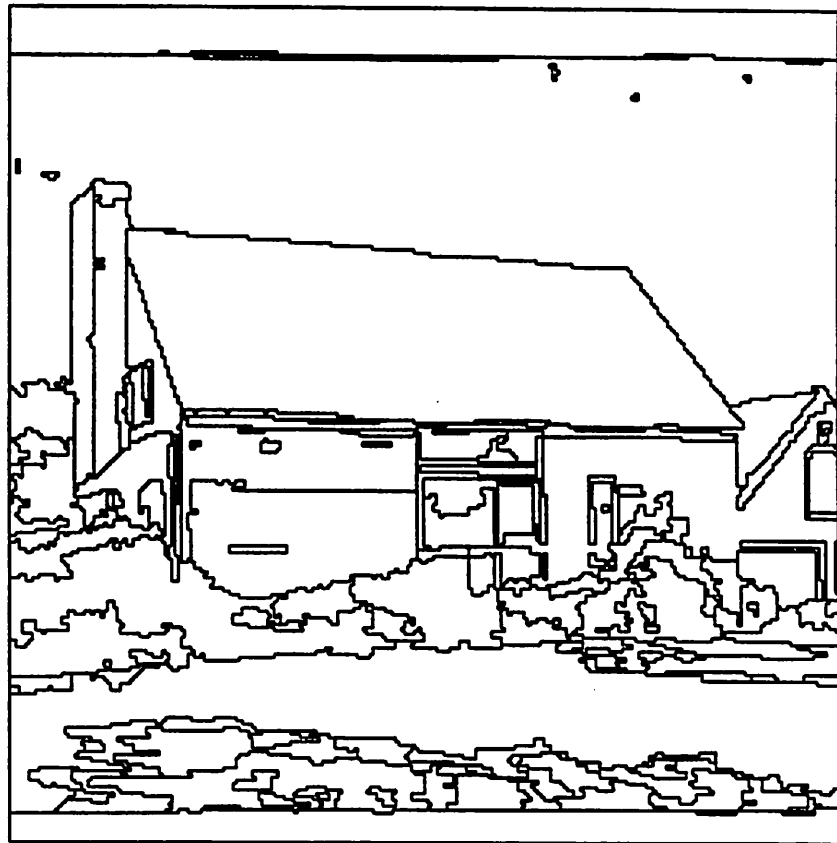
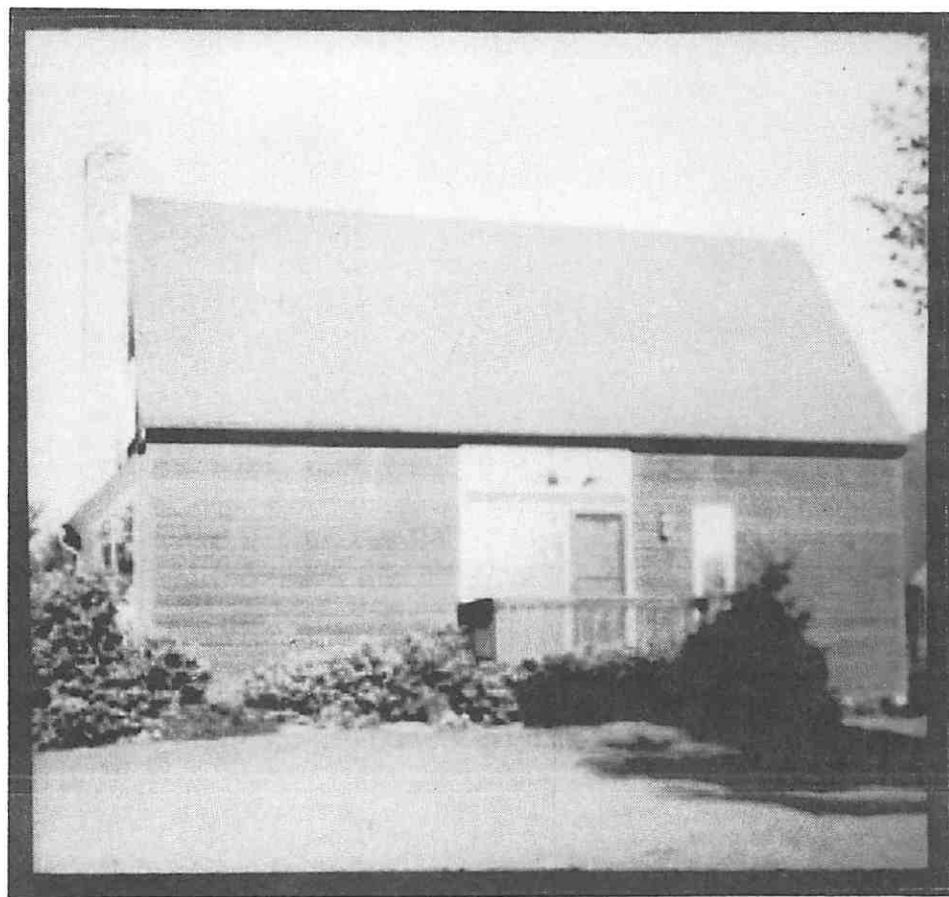


Figure 59 (continued).



(d)

Figure 59 (continued).

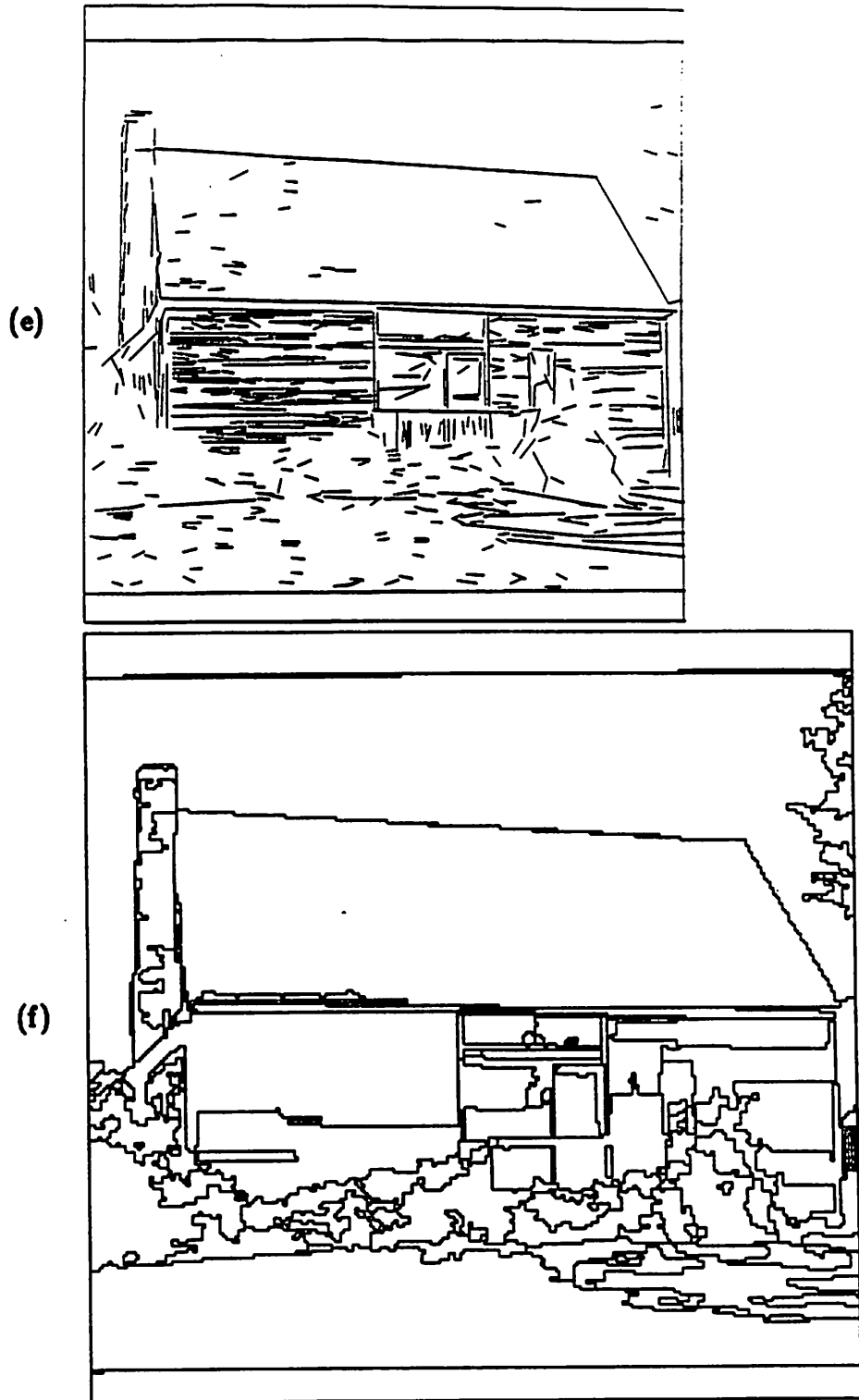
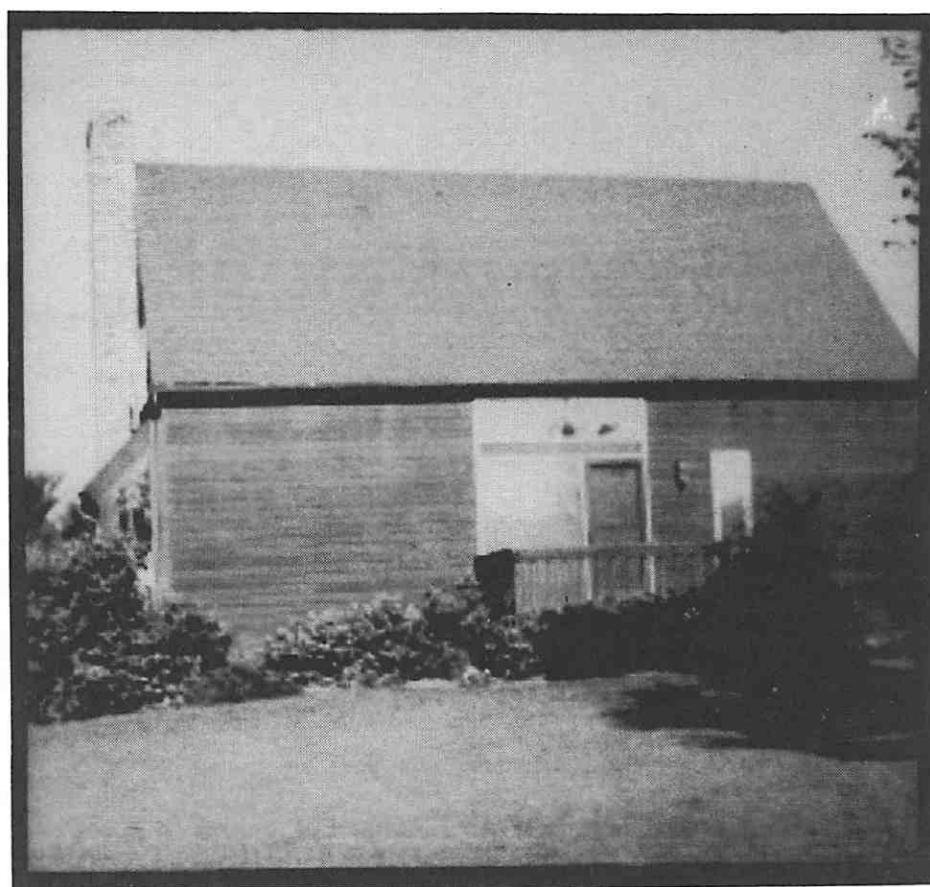


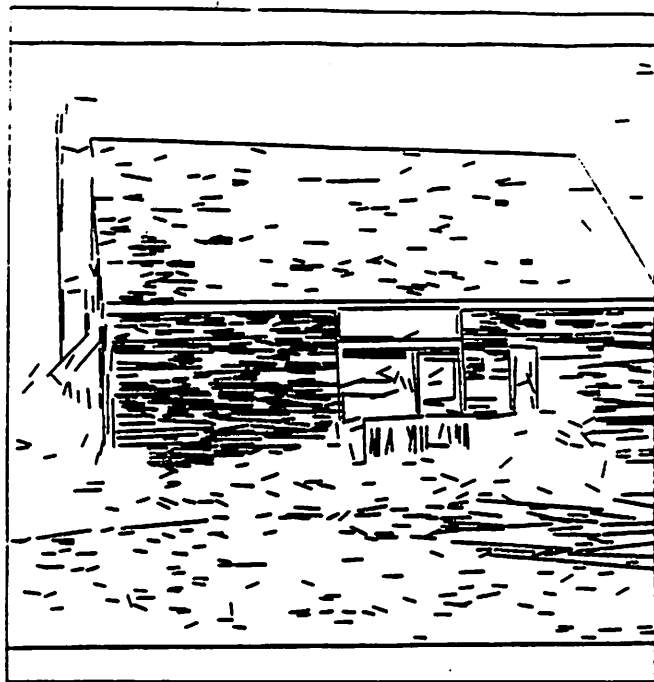
Figure 59 (continued).



(g)

Figure 59 (continued).

(h)



(i)

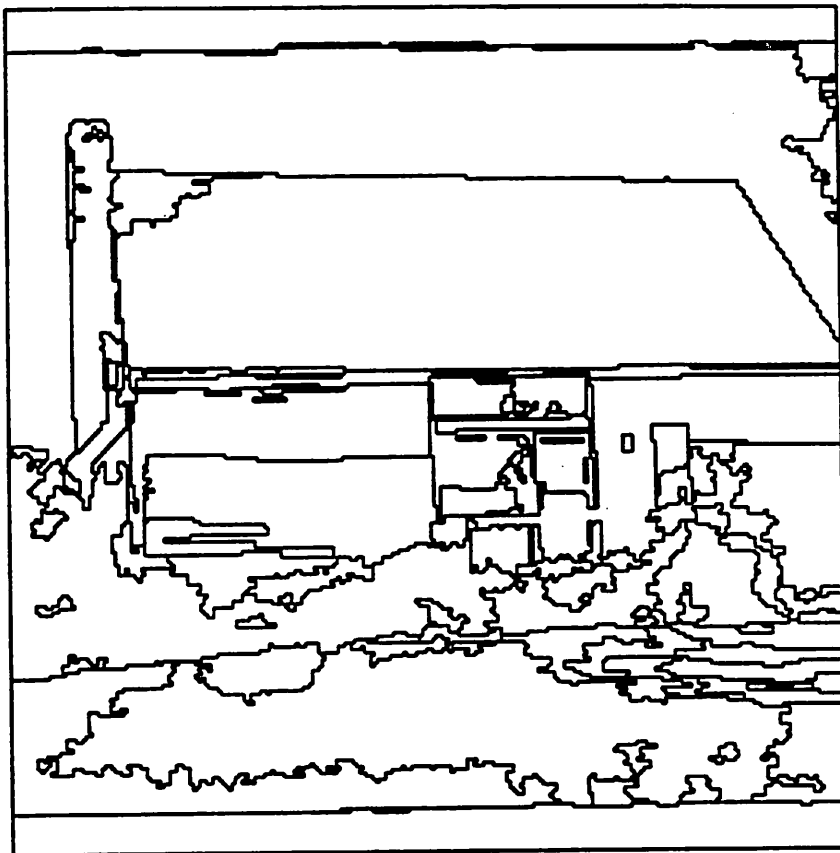


Figure 59 (continued).

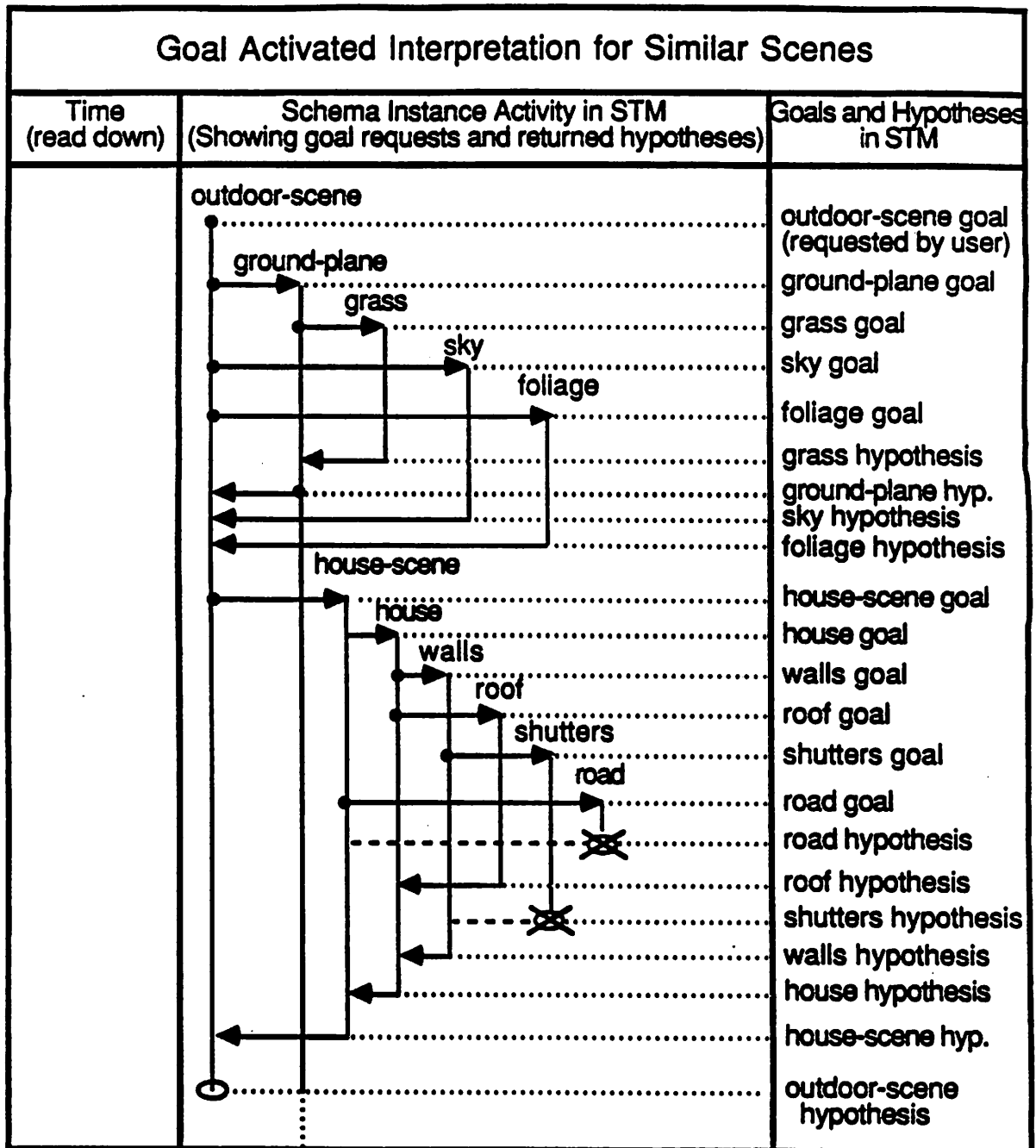
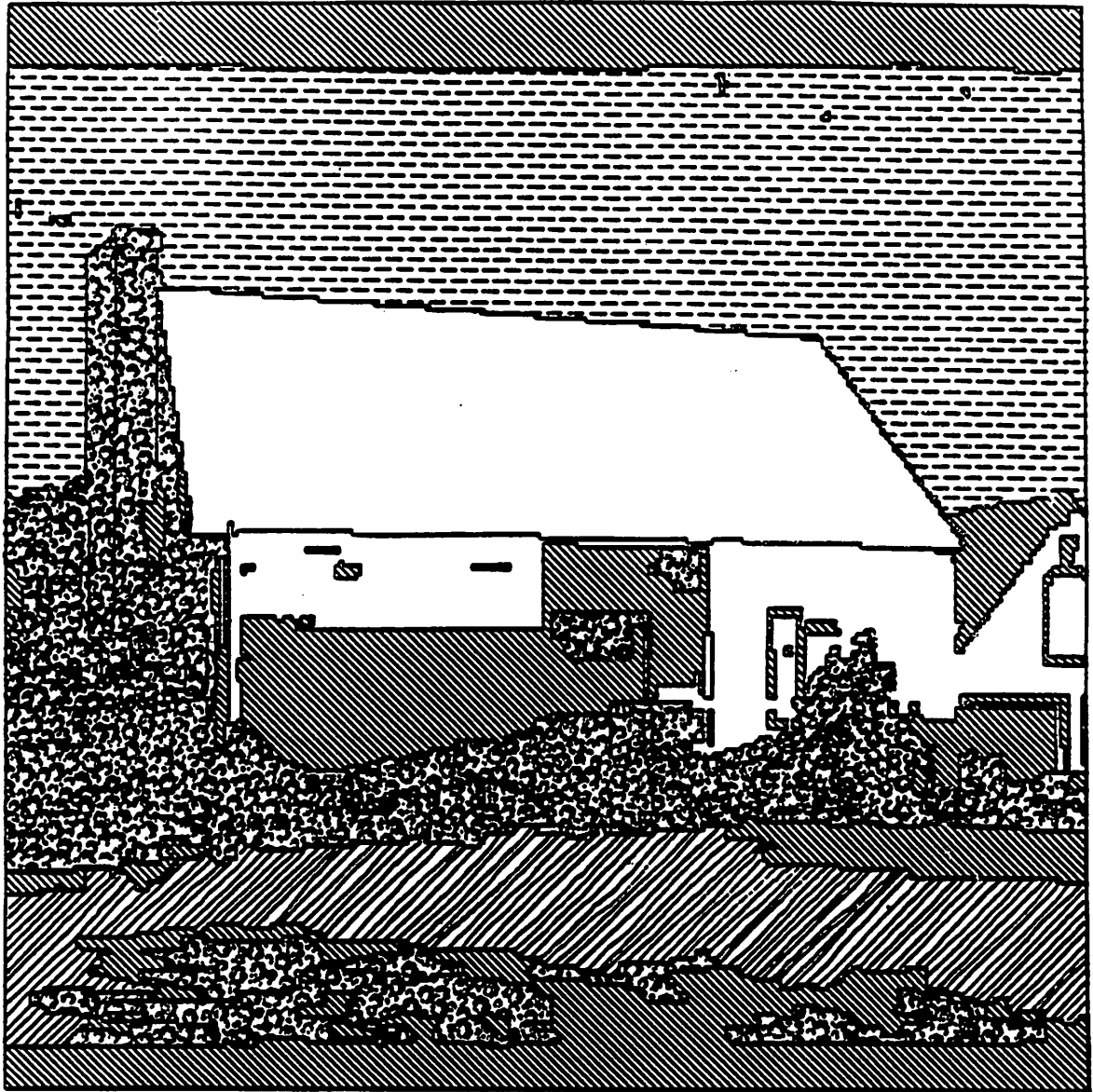


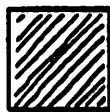
Figure 60. Activity Trace for Similar Images
 The activity traces of these interpretations differ only in small details of timing; the overall patterns of request and response are identical.

Figure 61. Interpretations of Similar Images

Differing views of the same scene give interpretations which differ in the details of object position and in image labeling. Furthermore, this example exhibits problems with the interpretation strategies as discussed in the text.



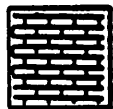
unlabeled



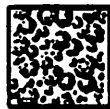
grass



sky



road

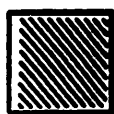
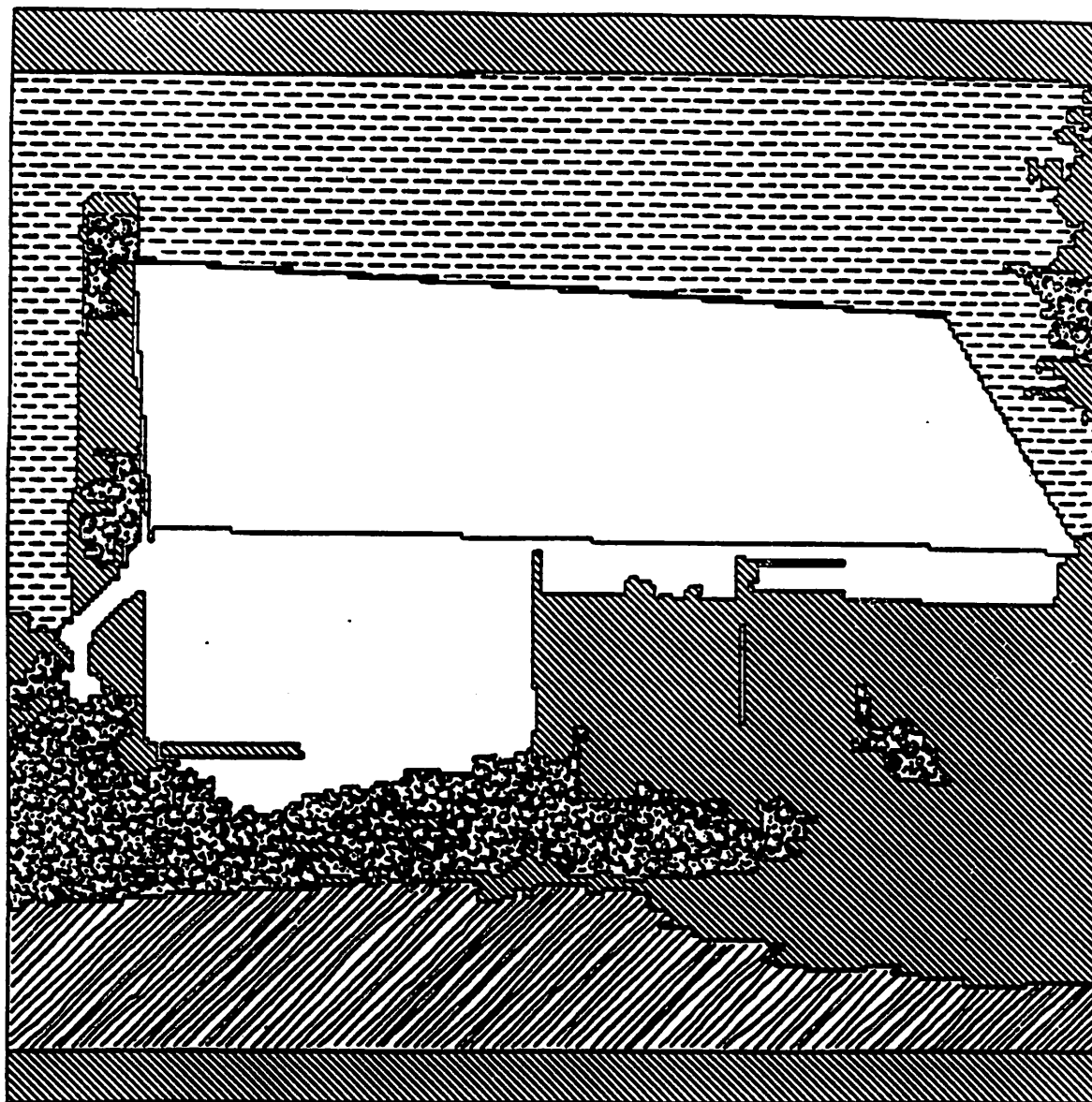


follage



house area

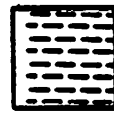
Figure 61.



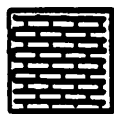
unlabeled



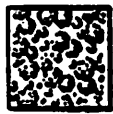
grass



sky



road



foliage



house area

Figure 61 (continued).

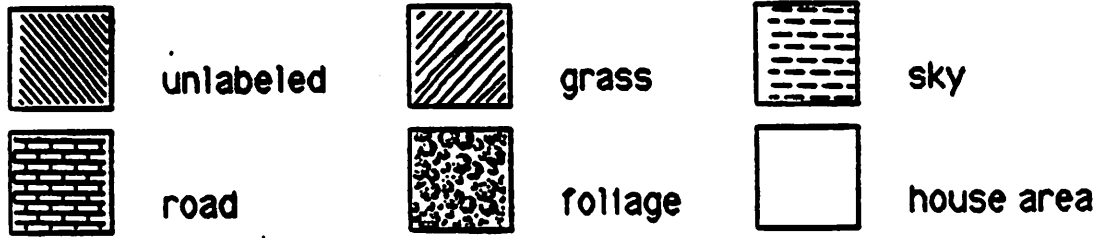
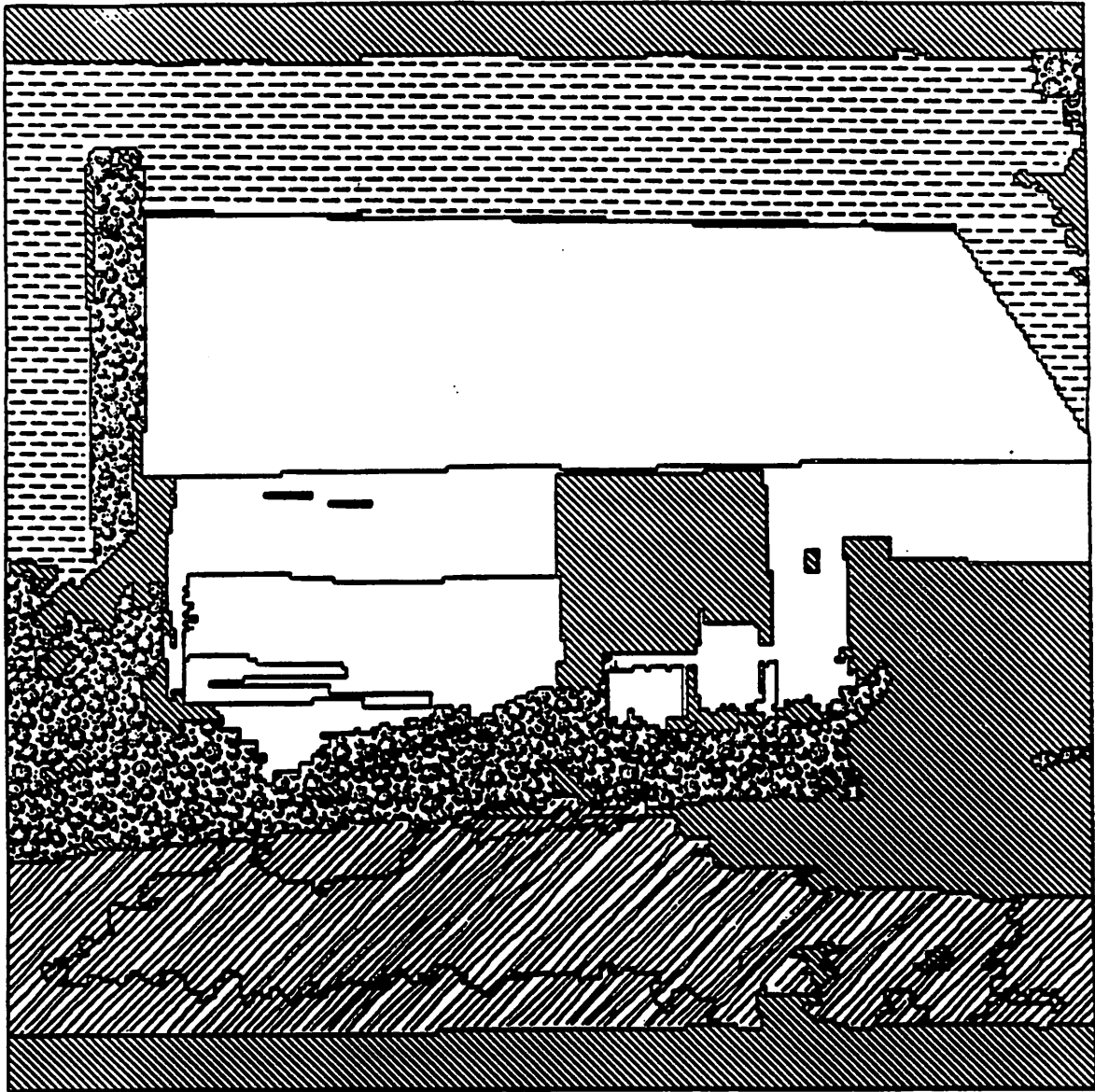


Figure 61 (continued).

in different labelings for the image areas associated with walls and foliage. Even the sky regions, though consistently labeled, change shape and location due to the differences in viewpoint. All these factors combine to produce markedly dissimilar labelings of the images.

Related to the differences in the image labeling produced by the projection of the interpretation network, there are differences within the network in the details of object placement associated with each object. Although the same relational network is produced for each interpretation, the spatial descriptions of the objects vary. For example, although the general description of the roof as a particular rectangle in space is similar for each of these images, the placement of the corners of the roof (as derived from each image) changes from image to image.

These interpretations differ from the one illustrated in Section 2. The lack of shutters and a road in the image causes the shutters and road instances to terminate without forming hypotheses, as shown in the trace of activation (Figure 60) by a circle-x pattern. Each strategy reports the lack of a hypothesis through the contract link of the goal and terminates without satisfying its activating goals.

In addition, the formation of the house geometry hypothesis for these interpretations fails. The reason for this failure illustrates the complexities in engineering knowledge for general objects. One of the tests used to verify the results from the roof hypothesis is the acceptability of the surface normal in the derived description of the roof. In the model of a house used for these interpretations the roof angle of the normal of the roof surface to a horizontal plane must be neither too small nor too large. This test is used to an advantage in eliminating bad hypotheses for roof, enabling the use of an alternate strategy, which we will discuss shortly. Unfortunately, when the limits on the acceptable range of angle values are made

too narrow, the roof in these images is rejected; and when they are made too wide, wrong interpretations from other images are accepted.

The reason the roof interpretation was not accepted as a starting point for the house geometry, in this case, was that the angle between the surface normal and the horizontal plane is too small (i.e., the plane of the roof is too upright). This error in interpretation of the roof results, in turn, from mismatches to the lines of the boundary of the roof which cause a shift in the image angle corresponding to the roof corner. These lines are mismatched because of the occlusion of the roof by the chimney (on the left), the boundary of which is close enough to the expected position of the roof boundary to be incorrectly interpreted as the edge of the roof.

When the roof interpretation strategy encounters the "out of bounds" surface normal it uses a second roof strategy, which (not surprisingly) finds the same roof area. This is returned to the house schema instance as the roof, but without the associated roof geometry information. Thus the house fails to construct the description of the house geometry.

More complete reasoning about the failure would be possible. For example, the rejected roof hypothesis is still available and its near match to the accepted roof hypothesis could be used to indicate that the geometric information was available. More basically, the tight coupling between the strategy to get the geometric information and the strategy that is doing the object recognition is a shortcoming that compounds the initial error made by the roof strategy. This argues strongly for more general strategies for handling geometric relations.

Despite the fact that this fragile relationship between the roof strategies and the method for constructing the geometric description of the house leads to a failure, the overall interpretation still remains fairly complete. Some details are mislabeled; specifically, the chimney, pieces of the house wall, and grass in shadow are all

labeled foliage. This is because the regions involved are reasonably dark, highly textured and lacking any other interpretation. The chimney is a case where there is some hope of being able to design an additional interpretation strategy based on context (such as the schemas for wires in the sky and the telephone pole illustrated in Chapter 3). Despite the fact that the interpretation for the overall structure of the house failed, some wall regions are identified by color consistency and the (image-based) spatial relations with the roof; but, more importantly, the house is recognized and placed in the image. Further, the interpretation of these similar images results in similar interpretation networks.

4.2 Detected Strategy Failure - Roof Interpretation

Our next example illustrates the use of the alternate interpretation strategy of the roof schema. It is a general illustration of two points: the decision to use parallel or serial activation, and the possible responses to the failure of an interpretation strategy. One possible means of creating a roof description would be to have several roof interpretation strategies active at the same time. There would then have to be some method of deciding which of the resulting, competing descriptions was best. However, because both of our roof strategies were fairly complex procedures, expensive in terms of processing time, we chose to apply first the strategy which experience had proven successful in the most general case; we would only use the other strategy when this one failed.

In the interpretation of the scene shown in Figure 62 the overall pattern of schema activation and hypothesis creation is identical to those shown in the previous examples: that is, the outdoor scene schema is activated by user request and its interpretation strategy activates the sky, ground-plane, and foliage schemas. When those three return results, the outdoor scene interpretation strategy activates the house-scene schema. The house-scene interpretation strategy produces (through the activation of the house schema) a house interpretation which gets incorporated

Figure 62. Occluded Roof Example

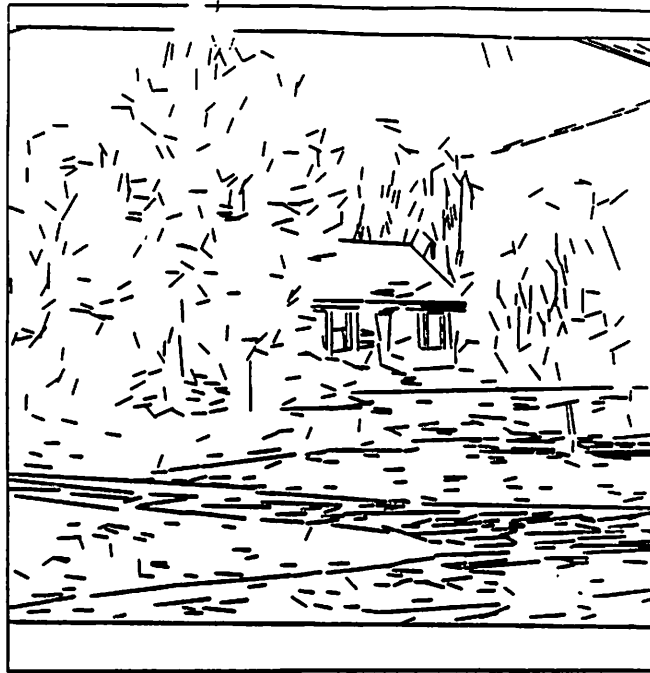
This is the image of the segmentation of a scene in which the roof is occluded. We use it to illustrate an interpretation in which the alternate roof strategy was used. (a) The digitized data; (b) the line data; (c) the region segmentation.



(a)

Figure 62.

(b)



(c)

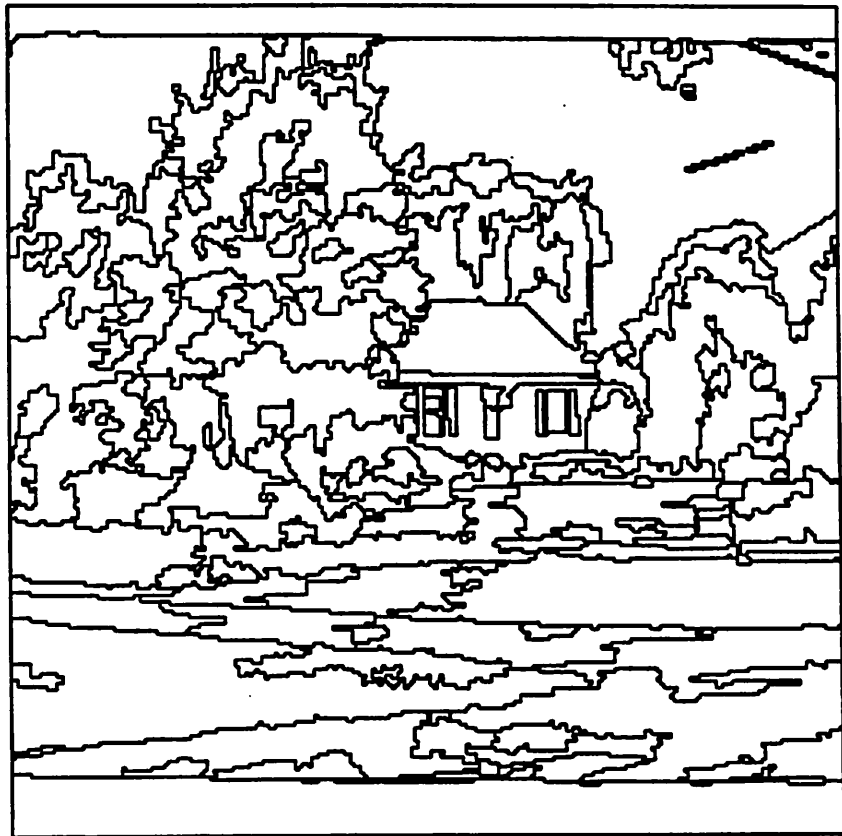


Figure 62 (continued).

into the description of the outdoor scene. What is striking in this example is the variation in the requirements for the roof strategy. The roof in this image is partly occluded, causing the eventual use of the second roof strategy.

The first roof strategy relies on finding evidence for all four sides of the roof rectangle in the image, proceeding blindly on the assumption that each of the four sides is present and that any evidence for the placement of a side is acceptable. When all four sides are present in the image, the geometry of the roof can be determined from the assumption that the roof is a rectangle and from the image information. In this image, forcing the interpretation through under the assumption that all four sides exist produces an interpretation of the roof based on connecting the visible fragments of the top and bottom of the roof at the endpoints of the visible portions. This leads to a roof hypothesis which is incorrect. Figure 63 shows the roof hypothesis developed by applying the first strategy prior to verification. Its rejection results from the size and surface normal being outside the limits set by the roof model. In such a case, a more general but less complete interpretation strategy must be used. The first one, which depends on finding all the edges of the roof, will not suffice. Thus, the second strategy is applied.

The alternate strategy searches for a match of some of the parts of the roof: in particular, finding three sides of the roof. The location of three sides, together with the knowledge of right angle corners, is sufficient to derive the surface orientation of the roof. The position of the fourth side comes from the default value in the LTM roof description for the ratio of the lengths of the sides of the roof rectangle. A roof is assumed to be (as a default) twice as long as it is high. This permits the strategy to supply a default position of the occluded roof edge, thus completing the description of the rectangle. As with our previous example the second strategy is unable to give an account of the complete (three-dimensional) geometry of the roof in space; hence, there is no hypothesis for the geometric structure of the house.

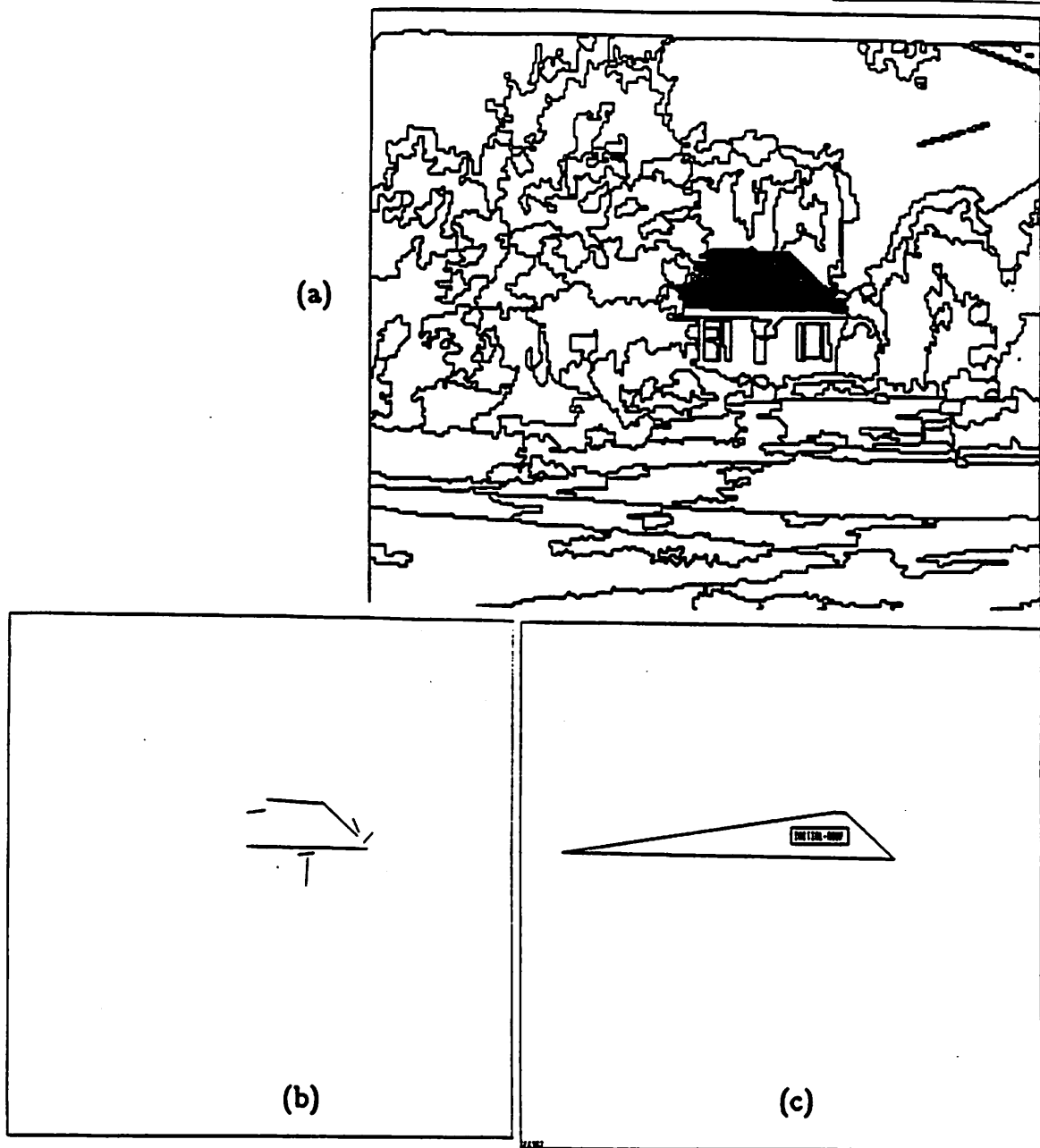


Figure 63. Incorrect Interpretation of Roof

The blind application of the first interpretation strategy for the roof leads to an incorrect interpretation that is rejected by a verification test based on size and surface normal. (a) The initial region chosen for the application of the first interpretation strategy. Note the absence of the fourth side. (b) The lines picked for boundary lines to complete the projection of a rectangle; (c) The projection of the hypothesized roof rectangle. The rectangle extends off into space from the viewer almost horizontally – obviously a bad hypothesis for this model of a house and roof.

Then interpretation proceeds as before, with the roof hypothesis being reported to the house instance and subsequently being used in the interpretation of house (see Figure 64). Thus, in this example, we see how an alternate interpretation strategy can be used to overcome possible problems in the interpretation of an object.

More important than the illustration of a second interpretation strategy, however, is the concept of specialized strategies for particular cases (such as viewpoint-specific interpretations) versus more general strategies. While general strategies may be more robust, especially if they use matching techniques of a rich descriptive structure in LTM, unfortunately they are not always useful; more general methods make the problems of engineering interpretation knowledge (procedures and integrated fine-grained knowledge) more difficult. A reasonable compromise is to attempt the development of procedures which apply specialized knowledge across as wide a range of conditions as possible, but are not useful for all images or all objects, and then to combine these with methods for determining when they are applicable. The idea of alternate specialized strategies is a step in that direction.

4.3 Viewpoint Differentiation and Viewpoint-Specific Strategies

This final example illustrates the use of viewpoint-dependent information being combined with a method for determining which view is actually present in the image. When the viewpoint, or general class of viewpoints, can be determined, then alternate strategy paths or (in general) alternate strategies can be applied based on the availability of viewpoint-specific information. In objects that are commonly viewed from one of several typical viewpoints, if those viewpoints can be easily distinguished (based perhaps on image data or prior interpretations), then the interpretation strategies can use viewpoint specific knowledge to further the interpretation. In fact, this type of knowledge has been used in all the interpretation results based on the house model shown thus far. Each of these interpretations assumed that the roof of the house was visible and that both the wall under the

Figure 64. Interpretation of Occluded Roof Example

The interpretation of the roof is shown. The second roof interpretation strategy utilizes three sides of the roof with an assumption of right angles. (a) The roof interpretation from the second interpretation strategy as projected onto the image plane; (b) the resulting labeling of the image for the house for the interpretation. The labeling for this figure is indicated in a slightly different manner from those in the other interpretations, due to the failure of the interpretation strategy for foliage in this interpretation. Because the exemplar region picked by the foliage strategy was relatively light in intensity, all of the otherwise unidentified regions were labeled foliage (and many others besides). Since any label in conflict with foliage effectively takes precedence, this left the image with all the unidentified regions labeled foliage.

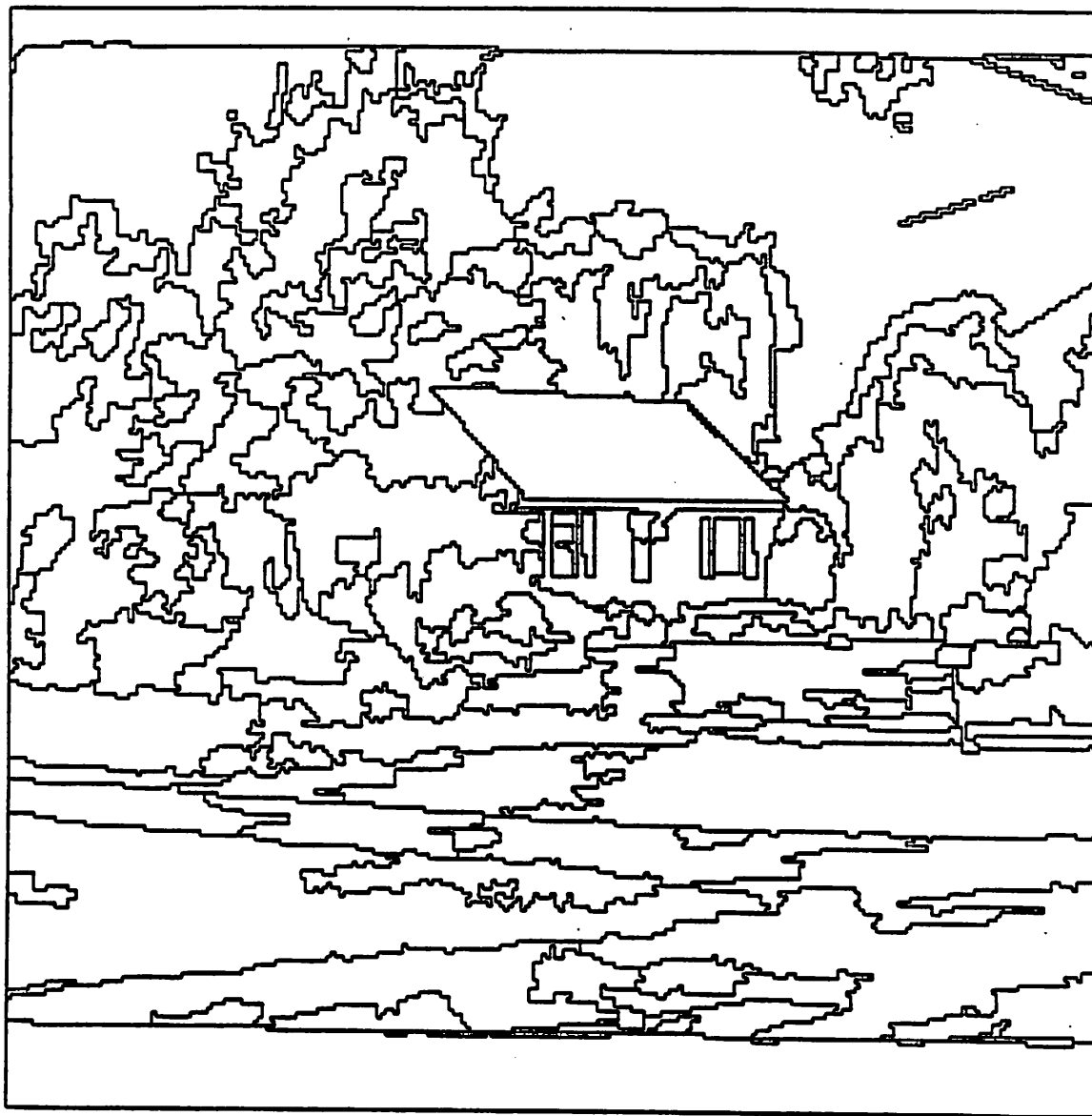
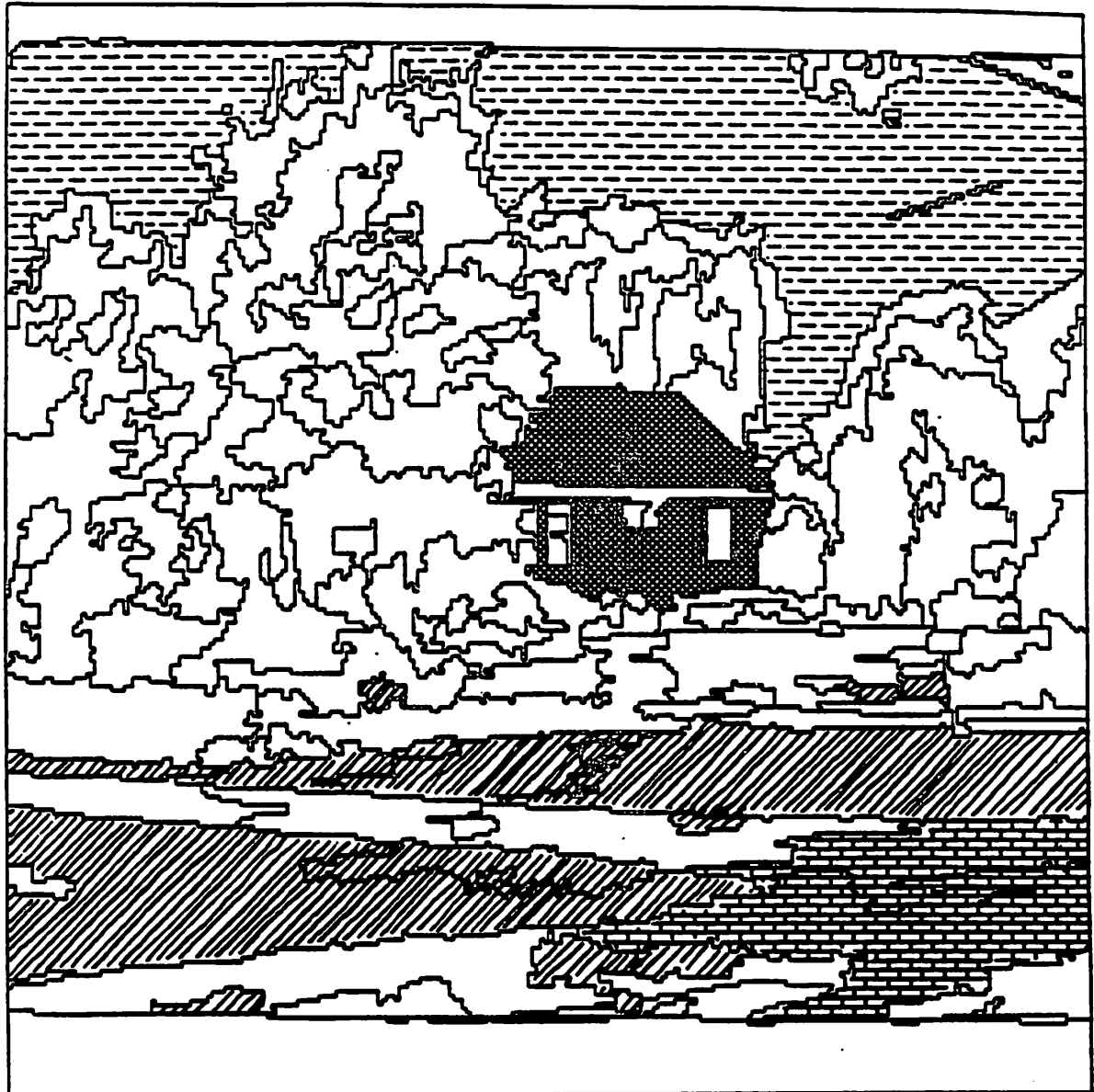
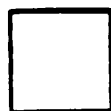


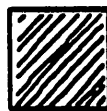
Figure 64.



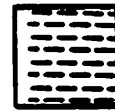
SEG307# 222--HOUSE (12-MAR-1985)



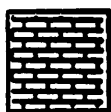
unlabeled



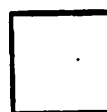
grass



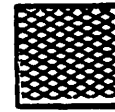
sky



road



foliage



house area

Figure 64 (continued).

eave of the roof (the "side" wall) and the wall adjacent to the peak of the roof (the "end" wall) were visible (or would be visible when the object parts were unoccluded). This is what we will call the "general view."

The basic concept of representing viewpoint-specific information was introduced in Chapter 2. There are three views associated with the house model used in this schema network. The interpretations of all the other images use a general view which combines the other two views: end-on and full-side. In this example, the interpretation is based on the end-on view. The image shown in Figure 65 presents a house from such a viewpoint. The interpretation of this image illustrates how the interpretation strategy for house can respond to a different viewpoint position.

In the interpretation of the house in this image, the house wall is found from the prior interpretation of the shutters by selecting and grouping regions of contrasting color to the shutters. In the attempt to interpret the roof no hypothesis can be formed. However, when the house interpretation strategy attempts to construct the house-geometry hypothesis, it discovers that the house presents an "end on" view. This discovery is derived from (image-based) information of the approximate shape of the interpreted portion of the wall, the angle to the "peak" of that wall being within an acceptable range for the view, and from the fact that the roof was uninterpreted. Rather than omit the geometric hypothesis (which normally follows from the failure of the roof schema), the house interpretation strategy constructs a partial description of the geometry with a hypothesis about where the non-visible house walls should be. The geometric description of the house is formed from the fitting of a symmetric pentagon to the boundary of the end wall; this forms a "cut out" of the end-wall. The interpretation strategy then adds a conjecture of the size and shape of the rest of the house using the default values from the model (the "end-wall" cutout is "extended" through space at right angles to the view plane, away from the viewer, for twice the width of the wall cutout). This hypothesis

Figure 65. End-on House Image

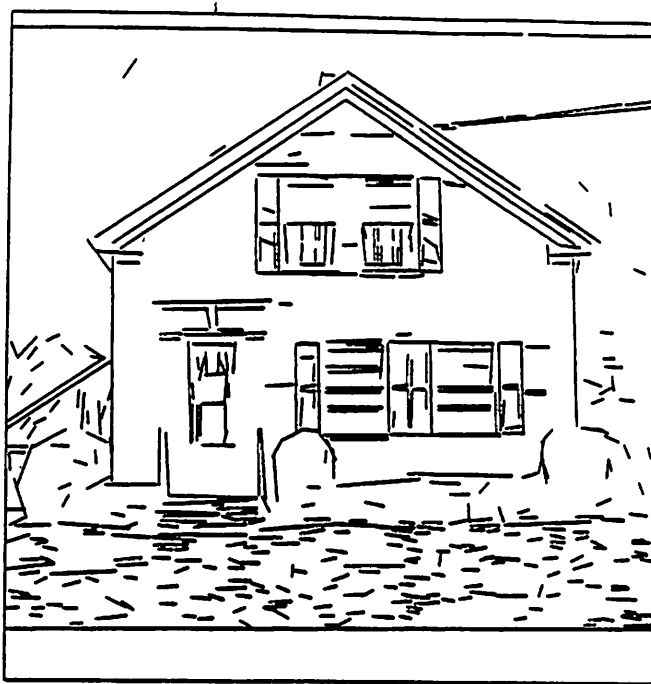
This image requires an alternate interpretation strategy for house, based on information about typical views. (a) The digitized data; (b) the line data; (c) the region segmentation.



(a)

Figure 65.

(b)



(c)

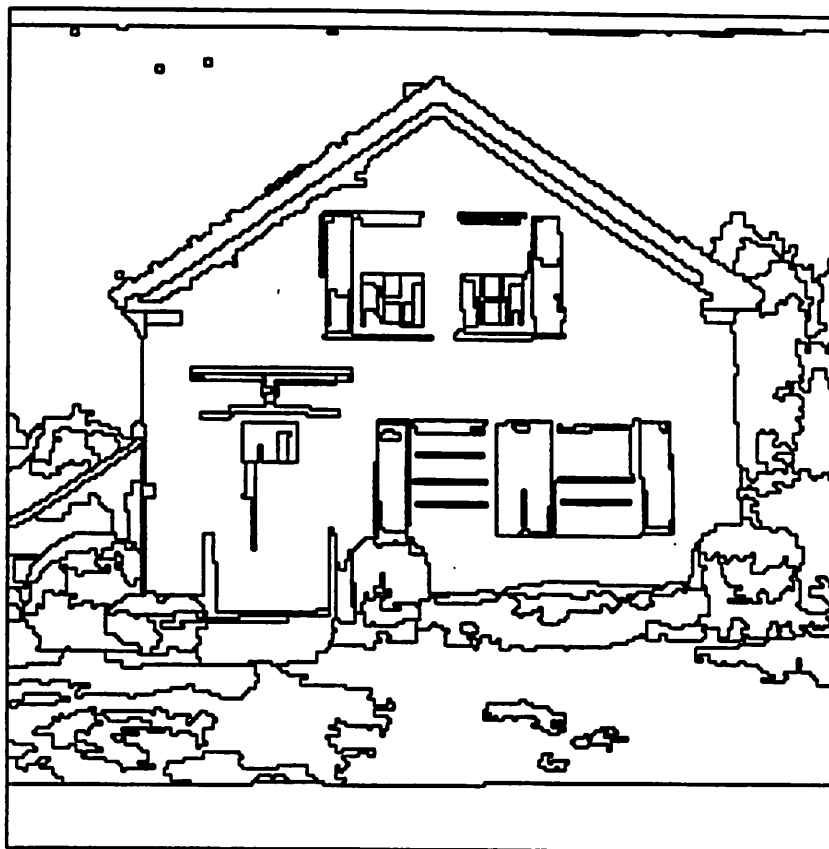


Figure 65 (continued).

is placed in STM. Seeing this description in the house-geometry hypothesis, the house-wall schema also recognizes the case of an end-on view and constructs the house-wall hypothesis accordingly (Figure 66).

From these examples we see that the current set of interpretation strategies can be used in combination to interpret additional images and that the basic structure of the interpretation network does not change over a wide range of scenes. Especially from the first example, we see that these methods might be used to keep track of objects in a slowly changing environment. This suggests that this interpretation system could be applied to dynamic scene understanding, where the use of incremental interpretation methods might make the overall problems of interpretation easier.

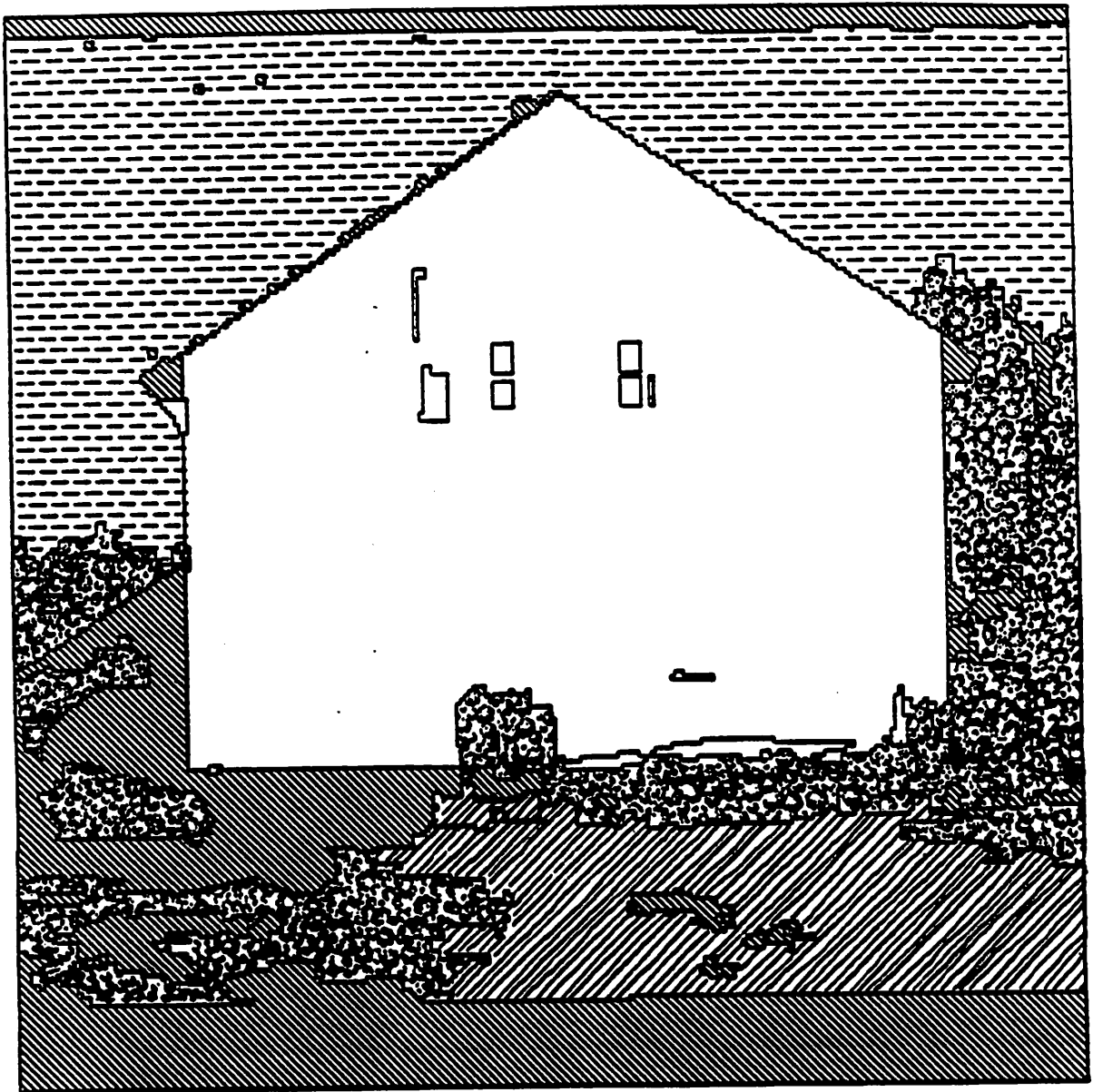
5. Pragmatic Design for Schema Development

Designing the schema network was a learning experience, from which we were able to distill some observations about design techniques. The integration of interpretation strategies in a schema network provides a mechanism for controlling the interpretation of a scene. Dependencies between and among schemas are expressed in the relational arcs and in the interpretation strategies. These dependencies, together with the results of partial interpretations, provide the basis for control of interpretation. This section outlines some heuristics for using that information.

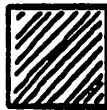
Deciding where to place information about object parts was one type of design decision. There were several places where we had to decide whether to represent object parts as schemas occurring separately or implicitly within the schema of the whole object. The guiding principle came from the number and type of dependencies among the parts. In objects where there were complex relationships among the parts, the interpretation strategy for the object created the hypotheses for the parts. Thus, the creation of shutter and shutter-pair hypotheses is the dominion

Figure 66. End-on House Interpretation

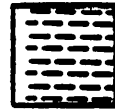
(a) The image labeling that results for the projection of the interpretation. (b) The house geometry hypothesis that results from the interpretation; details are in the text.



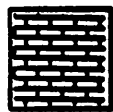
unlabeled



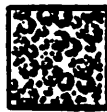
grass



sky



road



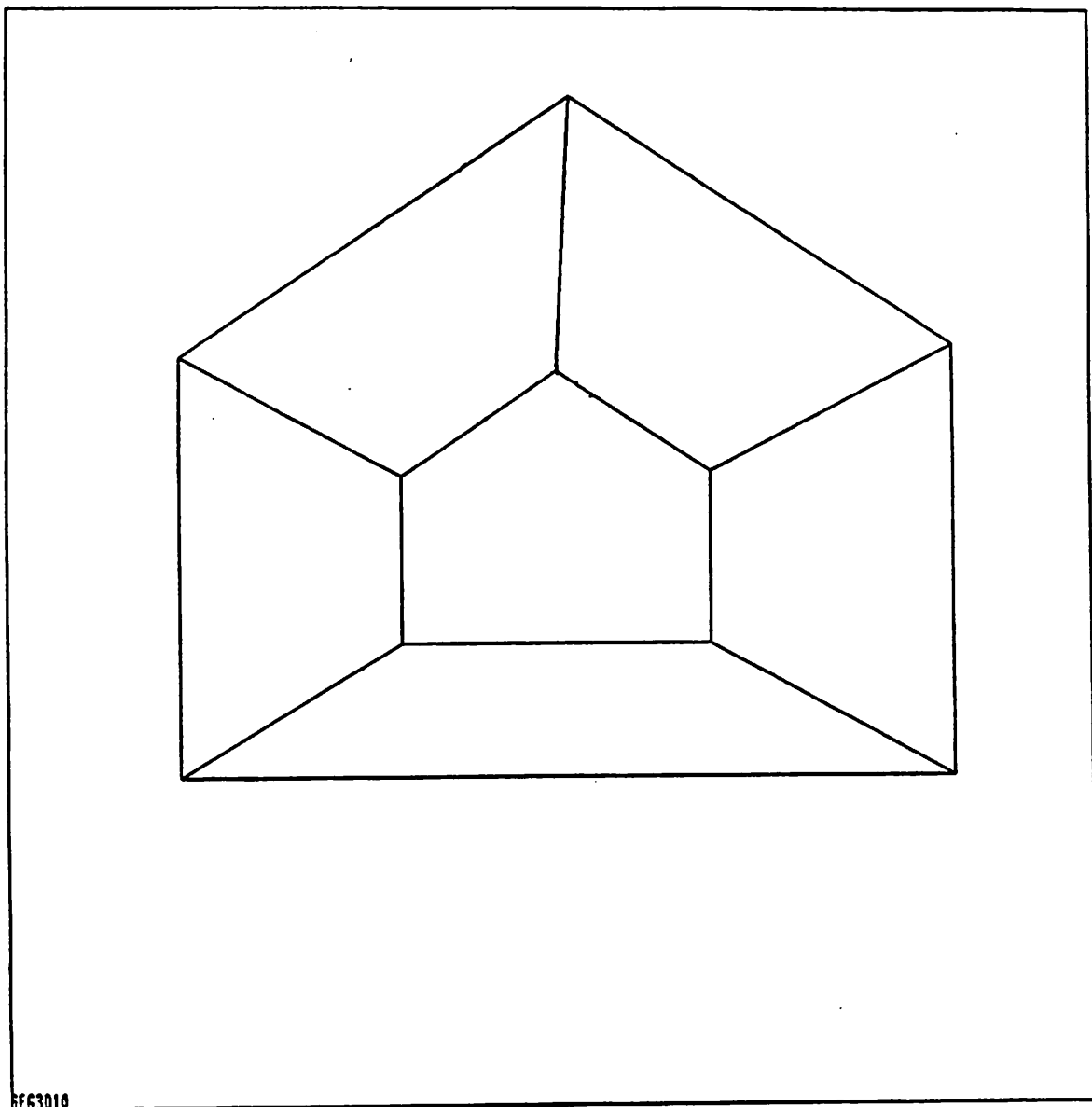
follage



house area

(a)

Figure 66.



(b)
Figure 66 (continued).

of the shutters schema, and the creation of the hypothesis for each individual wall of the house that of the walls schema. On the other hand, for a schema in which there were methods for recognizing the parts which were less dependent on the relations between the parts (i.e., roof as part of house), we chose to represent the parts with separate schemas. This was especially the case when the recognition and interpretation of those parts were more dependent on the data and the relations of the object (or object part) to the data, than with its relations with other parts. For example, the roof is part of the house and is handled by a separate schema.

Another type of design decision was determining how to handle the communication of information. This issue involves several considerations: When should the activation of schemas be made explicit and when should activation occur indirectly through STM? When should explicitly activated schemas be activated in parallel and when should they be activated in series? How should interpretation strategies be designed to maximize the amount of overlap in processing? Since direct communication was designed as part of explicit activation, the utilization of direct communication implies a direct control relationship between one schema instance and another. Indirect communication is also tied to a style of control relations where schemas can indirectly influence the actions of other schemas through the creation of hypotheses in STM. In this way, communication considerations are linked with the appropriateness of differing control styles.

When objects are closely related - such as the house, walls, and roof - the amount of influence they (potentially) have on one another suggests that a direct communication link will be helpful; and in fact, our experience confirms that direct communication and control (that is, through requesting a goal and communicating through the goal contract) should be done with objects that are closely related. For example, the interpretation of the house depends on the interpretation of its parts (roof and wall) and these are requested directly. Furthermore, the house

strategy suspends processing (by waiting for the part hypothesis) after making the goal requests. On the other hand, where the dependencies are less critical and more indirect (the objects on the ground-plane, for example) – especially when the related objects are not essential for further processing – then the interpretation strategy should do what processing it can and use the indirect communication methods to obtain information about hypotheses. If there is other useful processing, it can periodically test for the requested hypotheses in STM (i.e., poll STM), and when no other processing remains it can suspend activity (by waiting) until the requested hypotheses are posted in STM.

Our simple guiding principle on whether to wait or proceed is to proceed whenever possible. Processing should be suspended only when no further work can be done without the requested hypotheses. This follows directly from our interest in investigating parallel execution in interpretation. Allowing schema activity to proceed, whenever feasible, increases the opportunities for parallel execution and encourages (to some extent) the early acquisition of evidence to support primary hypotheses.

Another simple principle to enhance parallelism is to request all parts and subclasses early in the interpretation. This allows for branching to increase the number of schemas working on the image. One case where we did not do this is in the outdoor-scene interpretation strategy. The house-scene schema was not activated early. The schemas for sky, ground-plane, and foliage were requested first and processing was suspended until they returned their hypotheses because of the relative cost and payoff of activating those schemas as compared to the cost of activating the house-scene schema. Each of the schemas in the set that was activated early have simple interpretation strategies and are not complex objects. Furthermore, their combined interpretation allows the outdoor-scene schema to establish a context in which more specialized interpretation strategies can be invoked. This introduces, in

a straightforward way, the idea of processing cost, suggesting a direction for possible extension of control mechanisms.

As a last area of this discussion on design, we were also interested in how to handle the failure of interpretation strategies. Although we did not investigate these issues deeply, we do have a few suggestions. Ideally, interpretation strategies should be robust and adaptive, taking into account all the knowledge about an object and reasoning about missing data and incorrect interpretations to form the best interpretation. However, this ideal is far from realization. Much more needs to be understood about the types of interpretation strategies and methods for representing knowledge. In the interpretations shown here, our response to the failure of an interpretation strategy was to design alternate strategies. Sometimes the two strategies were sufficiently complementary to be run in succession. For example, our second roof interpretation strategy requires less information but is a method of constructing an object description that is less reliable. Thus, it was only invoked when the other strategy failed.

More frequently, as new interpretation strategies were incrementally improved, they replaced older ones. Further, as they required new knowledge, schemas were redesigned. Several generations of investigation and redesign contributed to the interpretation strategies and the schema network shown in this research. In the spirit of development in expert systems, we would design an interpretation strategy, run it on an image and redesign it until it worked. Then we would run it on another image and redesign it until it worked on both images. Then the process would be repeated for a small set of images. Finally the interpretation strategy would be combined with others and tested. The failures shown in the last section are but another step in this process. The effort of incremental design was aided by the design decisions that led to schemas with little interaction. A reasonable degree of modularity was achieved by basing schemas on objects and combining

all the interpretation tasks which interacted closely (as shutter-pairs) within one schema. Such modularity made the task of incremental knowledge engineering easier. Unfortunately, although we are all experts in seeing, expert knowledge is elusive; the development of an explicit knowledge base for vision is a protracted process.

CHAPTER V

CONCLUSION

In this chapter we will review the work presented in this dissertation, discuss the directions in which the current system might be extended, and summarize our contribution.

1. Review of This Work

This dissertation has described an experimental endeavor to investigate image interpretation. We have developed a system in which active processing elements work in parallel to construct a description of a scene. The basic structural element of our system is the schema which represents both procedures for recognition of an object and the description of the object, object-part, or scene.

The schema network, a set of schemas connected by relational arcs, is the basis for both control and representation in our system. For the interpretation of a particular set of scenes, the schemas in the network describe the classes of recognizable objects appearing in the scenes and the classes of scenes. Each type of schema refers to one or more interpretation strategies which control the ongoing interpretation. By encoding methods for recognition of objects, the interpretation strategies select which types of regions and lines correspond to the object and determine how those image-based primitives should be grouped in order to form a description of the object. The interpretation strategies construct the object and scene hypotheses and combine them into a network describing the scene, e.g., the interpretation network.

The schema network, through the collective actions of the interpretation strategies, constructs a description of the scene depicted in an image. Each interpretation

strategy, tied as it is to a schema instance, performs part of the interpretation process. Some interpretation strategies extract and group data, constructing a description of an object. Others request, collect, and group hypotheses, adding relational information to construct object or scene hypotheses. Schema instances tie the general descriptions in the schemas to both the processes performing the interpretation and to the interpretation network being constructed to describe the image. Each schema instance contributes the hypotheses for the type of object, object part, or scene represented by the schema. These hypotheses are combined by schema instances for the collective entity; schema instances for objects with parts construct descriptions from the hypotheses for their parts; the hypotheses for a scene are built by the scene schemas. Each such hypothesis is put into STM. Thus, each interpretation strategy builds portions of the interpretation network, while relying on the schema network for an indication of where the partial interpretations should be placed in the overall interpretation network.

The final result of the interpretation process, the interpretation network, expresses the rough position of objects in the scene and their part-to-whole relations. It can be viewed, by projection into the image plane, as an instance of the scene. Interpretations of images of a scene taken from similar viewpoints yield identical interpretations.

The experimental results in Chapter 4 show that each interpretation strategy has failings. When verification procedures to detect those errors are available, some of them can be overcome through the use of additional interpretation strategies. More importantly, however, these failures point out the need for increasingly general strategies which can deal flexibly with the problems of matching a general description to the image. Working against the evident need for flexible and general

methods, we find that the task of engineering the multifaceted fine-grained knowledge invites the use of specialized procedures for dealing with specific situations and for opportunistically using contextual information to limit complexity.

Our concentration on a modularized procedural representation aided in the development of interpretation strategies. We were often able to test a strategy in isolation before combining it with others. The clustering of information within a schema was also helpful in this regard, as was the tendency to construct interpretation strategies so that they started out with as little interaction with other interpretations as possible, relying on the verification stages to incorporate relational information. The limits of this approach were most acutely felt in the lack of sophistication in the type of communication and "coordination" available to the schema. When schema instances can only be linked to satisfy a goal, and then interact at very major stages in their respective strategies, they are necessarily limited in the types of cooperation that are possible.

The system described in Chapter 4 also deals with uncertainty in an *ad hoc* fashion. The problems of dealing with uncertainty for our limited domain are "finessed" by a direct reliance on redundancy of information and hardcoded expectations about the reliability of particular recognition strategies. We incorporate the "focus of attention" paradigm through the fine-grained control knowledge in the interpretation strategies by an implicit encoding of those image events which are more easily recognized and those object parts that are easier to identify. This method of control worked for some images, while for others it led directly to trouble. The central issue, one that requires much more research, is how to combine the rich and multifarious fine-grained knowledge for recognition of each individual object, together with the types of control information which we encoded in the interpretation strategies, in such a way as to deal with unanticipated errors, failures, and ambiguities.

2. Future Research

The richness of any research lies not only in the questions it answers but also in the questions that it causes us to ask. With our current system we attempted to increase our understanding of how to combine various types of knowledge about an object and about how an object can be recognized in an image. Our experience suggests several areas for future study, some of which we will discuss. In particular, we will briefly examine four areas of study: the building of additional interpretation strategies, the refinement of schemas and their methods of communicating, the use of richer descriptions of shape, and the use of pre-existent partial interpretations.

2.1 Developing Interpretation Strategies

The development of more sophisticated interpretation strategies is an important area of research. It is not difficult to argue that the results presented in Chapter 4 reveal interpretations that are incomplete as a result of missing schemas, or strategies that are not robust. We see that portions of each image are uninterpreted, some objects (clearly identifiable to a human observer) are not identified by the system, and of those objects identified much more could be said about the geometric structure and spatial relations. This is due, in part, to the extent of our exploration: we only designed schemas for a particular set of objects. Thus, one approach to improving the interpretation results is to continue the process of designing and coding schemas. We can use the general tactic suggested by the last set of interpretation strategies discussed in Chapter 3. That is, we could extend our repertoire of objects, but do so with the knowledge that their interpretation strategies can depend on other portions of the image having already been interpreted.

Insight into the techniques for developing interpretation strategies can also be gained by building schema networks for different domains, perhaps for a more limited domain. Since the investigation of how to structure object models for interpretation depends on both the domain of interpretation and the goals of the system, we

can gain a better understanding of the interpretation process by attempting interpretations in a variety of domains which require different sets of goals. Over time, this approach should lead to a deeper understanding of the types of interpretation strategies needed for a more general system.

Another issue in the investigation of interpretation strategies, the "grain size" of the schema, is tied to the second area of study. There is a trade-off between small and simple schemas that must depend more on interaction, and large and complex schemas that are relatively self-contained. The more complex a schema becomes, the more difficult it is to design robust and general interpretation strategies for it. Our house schema, for example, does not express a general notion of a house. As we explained in Chapter 2 we have modeled a very restricted notion of a house. Because of this, the schema would fail when given an image of a house of a different style.

We could respond to the problems of schema failure by saying that they are analogous to programming problems in which the programs do not account for all the cases; in order to improve an interpretation we include code for the missed cases. However, if we pursue this line of reasoning we encounter a problem. Interpretation systems tend to become large and complex, and a classical problem in programming methods is that large, complex systems are resistant to piecemeal debugging. One solution is to assure that the pieces (the schemas) are independent, as much as possible, from each other. This is analogous to the problem of determining a reasonable module size in the practice of modular programming; also, it is related to the task of defining the ideal "grain size" of a schema.

In the current implementation we chose to put all the knowledge "about an object" in a schema; however, we were somewhat arbitrary about what an object was. For example, we had a schema for a roof, while a shutter appears only as

part of a shutter-pair in the shutters schema. These decisions were based in part on the style of interaction possible among schema (more about this shortly). The decision was also based on the amount of processing needed to recognize each of the "objects." The interpretation strategies of each schema were approximately of the same size and complexity.

2.2 Communication Between Schemas

The second area of suggested study is the form of schemas and types of communications between their instances. The way in which schema instances interact helps determine the scope of any given schema. In our system, the amount of interaction between the processes creating hypotheses was a criterion as to whether an interpretation task was contained within a schema or split among several. If the process of creating an object hypothesis (e.g., the roof) was relatively independent, then both object and process were represented by a schema. However, where the interpretation of each individual object interacted closely with the interpretation of objects of the same type, then the construction of the hypotheses for all those objects was handled by the same schema. For example, the model used for shutters of a house required that they all be the same color and be oriented in the same way; further, their interpretation strategy depended on finding pairs. In this case, the shutter-pair model was embedded in the shutters schema. The interpretation of all shutters was produced by a single schema instance. With increased facility for communication between schemas, it would be possible to be less arbitrary about the scope of each schema type.

There is a trade-off between relying on large and complex schemas which do not require much communication and placing more complexity in the communications, and thereby working with small and simple schemas. Because of their relative lack of interaction, larger schemas permit greater modularity; however, this benefit is achieved at the cost of reduced flexibility. By corresponding more closely to isolated

visual events, smaller schemas would facilitate greater flexibility. Providing the schema instances with richer modes of communication would permit the creation of schemas for "smaller" groups of knowledge. For example, parallel lines occur as a structure in the roof, in the road, in the telephone pole, and in the shutters. Each such pair of lines would be described by different parameters, dependent on the particular object, but a general parallel line schema could be used to locate them based on the parameters (or, possibly, on ranges of parameters), especially if the object schema instance could communicate with the schema instance of the parallel line schema during the interpretation process. As an example, a message requesting information about the fitness of the current pair of parallel lines could cause the schema instance for the pair of lines to respond to such a request with information about how the data matched that pair of parallel lines, as well as to extend the pair, shift it slightly, or look for a different pair of parallel lines. This would require an extension of the current notion of a contract. Thus, we see that continued exploration in the interrelated directions of expanding methods for schema development, refining schema "size" or "granularity," and enriching communication among schema instances could lead to more general interpretation mechanisms.

2.3 Shape

Advances in the effective description of shape will undoubtedly affect the capabilities of interpretation strategies and facilitate the design of more general strategies. The description of three-dimensional object geometry is closely linked to the matching algorithms available, the image primitives extractable, and the types of two-dimensional shapes expressible in the interpretation system. Incorporating richer descriptions of shape into the interpretation strategies goes hand in hand with the concept of developing more general interpretation mechanisms. More general shape descriptions will lead to more general interpretation strategies, reducing the need for specialized interpretation strategies. Any interpretation strategy that

could be used for several types of objects, relying on the object description and applying general matching strategies, would be a reasonable base for a more robust set of schemas. This approach requires the development of object descriptions which are relevant to the matching processes that must deal with noisy image data and image feature extraction routines that are prone to errors; in other words, we need to develop descriptions of objects in which the primitives of representation *relate object characteristics to perception*, rather than to computer graphics or automated design. While much current research is developing methods for matching descriptions of geometric structure to a small class of man-made objects (i.e., machined parts), the description of natural objects awaits further study.

2.4 Use of Partial Interpretations

The final area of additional research suggested is the exploration of using a given partial interpretation of a scene. Rather than requiring the system to interpret the scene "from scratch," suppose that we start the interpretation with a given description. This approach would be useful in a robot workstation environment. We should not have to require the interpretation system to describe those parts of the scene that do not change from one picture to the next. It should be sufficient for the system to recognize that they match a description (perhaps from a former interpretation) that it already has stored in its STM. Note how such a description could come from a prior interpretation. Thus, if the interpretation system were required to interpret several views of a given scene, an initial interpretation could be used to supply predictions about what to expect in the other views of the scene; for example, the interpretation of one view of a house would permit the prediction of the colors of the walls from other views. With the addition of knowledge of how objects are expected to move in the environment, such incremental interpretation would also be useful in a motion interpretation system, as in a mobile robot. When a system is able to incorporate the information from previous or supplied

partial interpretations, it can make additions to the description of the scene and subsequently use the updated interpretation for further analysis.

3. General Contribution

This research has emphasized synthesis. Although the image segmentation routines were developed by others and standard techniques for organizing the data were used, we have successfully combined them into a working whole. The details of the interpretation strategies use many common techniques, such as matching 3D models of geometric structure to line segments in the image, using image feature measurements to identify classes of objects, and using a network describing object-part relations to guide the interpretation process. None of these methods is new. What is new is their combination into a system that works to a reasonable extent on images from a complex domain. A large range of knowledge has been integrated into a fine-grained representation of interpretation strategies that can successfully cooperate to build a global interpretation.

Another contribution is the idea of parallel, distributed control as developed and explored in the use of schemas. By taking advantage of redundancies in the image, in the scenes, and in the structures describing them, by using the potential for parallelism indicated by spatially separated image events, and by exploiting notions developed in other fields of AI for frame-like representations, we have developed and applied the concept of schemas for image interpretation. These schemas, operating in parallel and cooperating by communicating and sharing information, control complex interpretation processes that recover object identities from photographs of outdoor scenes.

Finally we have advanced, by a small step, in the direction of building a general computer vision system. By concentrating on methods that resulted in a working

system, we have examined the pragmatics of building an image interpretation system. The success that we have had here is a foundation for continuing exploration.

Because it is a complex problem, requiring the development of a deep understanding of both the visual process and of its relation to knowledge about the world, general machine vision will need to be approached in small steps. Image interpretation in any domain requires a large amount of knowledge about the particular domain, and the development of this knowledge bears similarity to the construction of "expert knowledge" systems: the information needs to be collected, checked for consistency, experimentally verified, modified, and compiled. The challenge lies in the fact that the domain of vision is not one in which there is a clear understanding of the expert knowledge; it is extremely difficult for anyone to give a description (especially at the level of detail required by an expert system) of the information that we use to see.

While we are waiting for the technology of expert systems to catch up to the needs of this domain, it behooves us to attempt to gain an understanding by incremental improvement on machines that see. Initially we interpret one image or the images in a restricted domain, then expand from this to a couple of similar images or a slightly more complex domain. By these small steps we will approach an understanding of the processes and the methods needed for a general, knowledge-based machine vision system.

B I B L I O G R A P H Y

- [AKI80] Akin, O., "Understanding as a Function of Image Compression," Technical Report CMU-CS-80-108, CMU, January 31, 1980.
- [ARB78] Arbib, M., "Segmentation, Schemas, and Cooperative Computation," in *Studies in Mathematical Biology, Part I* (S. Levin, ed.), MAA Studies in Mathematics, Vol. 15, 1978, pp. 118-155.
- [BAD79] Badler, N. I., O'Rourke, J. and Toltzis, H., "A Spherical Human Body Model for Visualizing Movement," *IEEE Proceedings*, Vol. 67, No. 10, October 1979, pp. 1397-1403.
- [BAJ78] Bajcsy, R. and Joshi, A., "A Partially Ordered World Model and Natural Outdoor Scenes," in *Computer Vision Systems* (A. Hanson and E. Riseman, eds.), Academic Press, 1978.
- [BAL82] Ballard, D. and Brown, C., *Computer Vision*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.
- [BAL78] Ballard, D., Brown, C. and Feldman, J., "An Approach to Knowledge-Directed Image Analysis," in *Computer Vision Systems* (A. Hanson and E. Riseman, eds.), Academic Press, 1978.
- [BAR71] Barrow, H. G. and Popplestone, R. J., "Relational Descriptions in Picture Processing," in *Machine Intelligence 6* (B. Meltzer and D. Michie, eds.), Edinburgh University Press, Edinburgh, 1971.
- [BAR76] Barrow, H. and Tenenbaum, J., "MSYS: A System for Reasoning About Scenes," Technical Note 121, AI Center, Stanford Research Institute, April 1976.

- [BAR32] Bartlett, F. C., *Remembering - A Study in Experimental and Social Psychology*, Cambridge University Press, London, 1932.
- [BIE81] Biederman, I., "On the Semantics of a Glance at a Scene," *Perceptual Organization* (M. Kubovy and J. R. Pomerantz, eds.), Erlbaum, 1981.
- [BIE82] Biederman, I., "Scene Perception: Detecting and Judging Objects Undergoing Relational Violations," *Cognitive Psychology*, Vol. 14, 1982, pp. 143-177.
- [BIE83] Biederman, I., "Scene Perception: A Failure to Find Benefit from Prior Expectancy or Familiarity," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 9, No. 3, 1983, pp. 411-429.
- [BIE73] Biederman, I., Glass, A., and Stacy, E. W., "Searching for Objects in Real-World Scenes," *Journal of Experimental Psychology*, Vol. 97, No. 1, 1973, pp. 22-27.
- [BIN82] Binford, T., "Survey of Model Based Image Analysis Systems," *International Journal of Robotics Research*, Vol. 1, No. 1, Spring 1982, pp. 18-64.
- [BLI77] Blinn, J. F., "Models of Light Reflection for Computer Generated Images," in *Proc. of SIGGRAPH-77* published as *Computer Graphics*, Vol. 11, No. 2, Summer 1977, pp. 192-198.
- [BRA76] Brachman, R., "What's in a Concept: Structural Foundations for Semantic Networks," BBN Report 3433, 1976.
- [BRA82] Brady, M., "Computational Approaches to Image Understanding," *Computing Surveys*, Vol. 14, 1982, pp. 3-71.
- [BRO81] Brooks, R., "Symbolic Reasoning Among 3-D Models and 2-D Images," Technical Report STAN-CS-81-861, Department of Computer Science, Stanford University, Stanford, California, June 1981.

- [BUR84] Burns, J. B., Hanson, A. R., and Riseman, E. M., "Extracting Straight Lines," in *Proc. Image Understanding Workshop*, Defense Technical Information Center, Alexandria, VA, 1984.
- [CON82] Conklin, E. J., *Localized Planning of Discourse Generation Using Saliency*, Ph. D. Dissertation, COINS Department, University of Massachusetts, November 1982.
- [COR81] Corkill, D. D., and Lesser, V. R., "A Goal-Directed Hearsay-II Architecture: Unifying Data- and Goal-Directed Control," Technical Report 81-15 COINS Department, University of Massachusetts, June 1981.
- [CRO82] Crow, F. C., "A More Flexible Image Generation Environment," in *Proc. SIGGRAPH '82* published as *Computer Graphics*, Vol. 16, No. 3, 1982, pp. 9-18.
- [DUD78] Duda, R. O., Hart, P. E., Nilsson, N. J., and Sutherland, G. L., "Semantic Network Representations in Rule-Based Inference Systems," in *Pattern-Directed Inference Systems* (Waterman and Hayes-Roth, eds.), Academic Press, New York, 1978.
- [ERM80] Erman, L., Hayes-Roth, F., Lesser, V. and Reddy, D., "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty," *Computing Surveys*, 12(2), June 1980, pp. 213-253.
- [FAL72] Falk, G., "Interpretation of Important Line Data as a Three-Dimensional Scene," *Artificial Intelligence*, Vol. 3, No. 1, Spring 1972, pp. 77-100.
- [FAU79] Faux, I. D., and Pratt, M. J., *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd., West Sussex, England, 1979.
- [FOL82] Foley, J. D., and VanDam, A., *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, Reading, Massachusetts, 1982.

[FUR81] Furth, H. G., *Piaget and Knowledge: Theoretical Foundations*, University of Chicago Press, Chicago, 1981.

[GLI82] Glicksman, J., "A Cooperative Scheme for Image Understanding Using Multiple Sources of Information," Ph. D. Dissertation, University of British Columbia, November 1982.

[GOF74] Goffman, E., *Frame Analysis, An Essay on the Organization of Experience*, Harper and Row, New York, 1974.

[GOL83] Goldberg, A., and Robson, D., *Smalltalk-80: The Language and Its Implementation*, Addison-Wesley, 1983.

[GRI85] Griffith, J., Personal Communication, University of Massachusetts, 1985.

[HAN78] Hanson, A. and Riseman, E., "VISIONS: A Computer System for Interpreting Scenes," in *Computer Vision Systems* (A. Hanson and E. Riseman, eds.), Academic Press, 1978, pp. 303-333.

[HAN83] Hanson, A. and Riseman, E., "A Summary of Image Understanding Research at the University of Massachusetts," Technical Report 83-35, COINS Department, University of Massachusetts, October, 1983.

[HAN85] Hanson, A., et.al. "A Methodology for the Development of General Knowledge-Based Vision Systems," Technical Report, COINS Department, University of Massachusetts, September 1985.

[HOR77] Horn, B., "Understanding Image Intensities," *Artificial Intelligence*, Vol. 8, No. 2, 1977, pp. 201-231.

[IKE80] Ikeuchi, K., "Numerical Shape for Shading and Occluding Contours in a Single View," AI Memo 566, AI Lab, MIT, February 1980.

- [KAW82] Kawaguchi, Y., "A Morphological Study of the Form of Nature," *Computer Graphics*, Vol. 16, No. 3, July 1982, pp. 223-232.
- [KOH83] Kohler, R., "Integrated Syntactic Knowledge for Segmentation," Ph. D. Dissertation, University of Massachusetts, September 1983.
- [LES77] Lesser, V. R., and Erman, L. D., "A Retrospective View of the Hearsay-II Architecture," *IJCAI-77*, August 1977, pp. 790-800.
- [LEV81] Levine, M. and Shaheen, S., "A Modular Computer Vision System for Picture Segmentation and Interpretation," *IEEE PAMI*, Vol. 3, No. 5, September 1981, pp. 540-556.
- [LOW78] Lowrance, J., "GRASPER 1.0 Reference Manual," Technical Report 78-20, COINS Department, University of Massachusetts, December 1978.
- [LOW82] Lowrance, J., "Dependency-Graph Models of Evidential Support," Ph. D. Dissertation, University of Massachusetts, 1982.
- [MAC78] Macworth, A., "Vision Research Strategy: Black Magic, Metaphors, Mechanisms, Miniworlds, and Maps," in *Computer Vision Systems* (A. Hanson and E. Riseman, eds.), Academic Press, New York, 1978.
- [MIN75] Minsky, M., "A Framework for Representing Knowledge," in *The Psychology of Computer Vision*, McGraw-Hill, 1975.
- [MIN80] Minsky, M., "The Society Theory of Thinking," in *Artificial Intelligence: An MIT Perspective* (Winston and Brown, eds.), The MIT Press, 1980.
- [NAG80] Nagao, M. and Matsuyama, T., *A Structural Analysis of Complex Aerial Photographs*, Plenum Press, New York, 1980.

- [NAG79] Nagin, P. A., "Studies in Image Segmentation Algorithms Based in Histogram Clustering and Relaxation," Ph. D. Dissertation available as Technical Report 79-15, COINS Department, University of Massachusetts, September 1979.
- [OHT80] Ohta, Y., "A Region-Oriented Image-Analysis System by Computer," Ph. D. Dissertation, Kyoto University, Department of Information Science, Kyoto, Japan, 1980.
- [OVE84] Overton, K. J., "The Acquisition, Processing, and Use of Tactile Sensory Data in Robot Control," Ph. D. Dissertation available as Technical Report 84-08, COINS Department, University of Massachusetts, May 1984.
- [PAR80] Parma, C., Hanson, A. and Riseman, E., "Experiments in Schema-Driven Interpretation of a Natural Scene," Technical Report 80-10, COINS Department, University of Massachusetts, 1980. Also in *NATO Advanced Study Institute on Digital Image Processing* (R. Haralick and J. C. Simon, eds.), Bonas, France, 1980.
- [PAU77] Paul, J. L., "An Image Interpretation System," Ph. D. Dissertation, University of Sussex, June 1977.
- [PEN83] Pentland, A. P., "Fractal-Based Description," in *Proc. IJCAI-83*, Karlsruhe, West Germany, 1983.
- [RAN75] Random House, *The Random House College Dictionary*, 1975.
- [REV83] Reeves, W. T., "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects," *ACM Transactions on Graphics*, Vol. 2, No. 2, April 1983, pp. 91-108.
- [REY84] Reynolds, G., et al., "Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation," *Proc. of the Work-*

shop on Computer Vision: Representation and Control, Annapolis, Maryland, April 30-May 2, 1984, pp. 238-247.

[ROB64] Roberts, L. G., "Machine Perception of Three-Dimensional Solids," *Symposium of Optical and Electro-optical Information Processing Technology*, Boston, 1964, pp. 159-197.

[ROG76] Rogers, D. F., and Adams, J. A., *Mathematical Elements for Computer Graphics*, McGraw-Hill, New York, 1976.

[SHA76] Shafer, G., *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

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

[SHO76] Shortliffe, E. H., *Computer-Based Medical Consultations: MYCIN*, North-Holland, New York, 1976.

[SLO77] Sloan, K., "World Model Driven Recognition of Natural Scenes," Ph. D. Dissertation, Moore School of Electrical Engineering, University of Pennsylvania, 1977.

[SMI84] Smith, A. R., "Plants, Fractals, and Formal Languages," *Computer Graphics*, 18, 3, July 1984, pp. 1-10.

[TEN76] Tenenbaum, J. M. and Barrow, H., "Experiments in Interpretation-Guided Segmentation," Technical Note 123, AI Center, Stanford Research Institute, 1976; also in *AI Journal*, Vol. 8, No. 3, 1977, pp. 241-274.

[TSO84] Tsotsos, J. K., "Representational Axes and Temporal Cooperative Processes," Technical Report RCBV-TR-84-2, University of Toronto, April 1984.

- [TUR74] Turner, K. J., "Computer Perception of Curved Objects Using a Television Camera," Ph. D. Dissertation, University of Edinburgh, Edinburgh, 1974.
- [VOE78] Voelcker, H. B., et. al., "The PADL-1.0/2 System for Defining and Displaying Solid Objects," *Computer Graphics* 12, 3, August 1978, pp. 257-263.
- [WES82] Wesley, L., and Hanson, A., "The Use of an Evidential-Based Model for Representing Knowledge and Reasoning about Images in the VISIONS System," *Proc. of the Workshop on Computer Vision*, Rindge, New Hampshire, August 1982, pp. 14-25.
- [WES83] Wesley, L., "Reasoning About Control: The Investigation of an Evidential Approach," *IJCAI-83*, August 1983, pp. 203-206.
- [WES86] Wesley, L., "The Application of an Evidential Based Technology to a High-Level Knowledge-Based Image Interpretation System," Ph. D. Dissertation, University of Massachusetts, in preparation (expected in 1986).
- [WEY83] Weymouth, T. E., Griffith, J. S., Hanson, A. R., Riseman, E. M., "Rule Based Strategies for Image Interpretation," *Proc. AAAI-83*, August 1983, pp. 429-432; available in a longer version in *Proc. of the DARPA Image Understanding Workshop*, Arlington, VA, June 1983, pp. 193-202.
- [WIL81] Williams, T. D., "Computer Interpretation of a Dynamic Image from a Moving Vehicle," Ph. D. Dissertation, available as COINS Technical Report 81-22, University of Massachusetts, Amherst, May 1981.
- [YOR80] York, B., Hanson, A. and Riseman, E., "3D Object Representation and Matching with B-Splines and Surface Patches," *Proc. IJCAI-7*, August 1980, pp. 648-651.

[YOR81] York, B., "Shape Representation in Computer Vision," COINS Technical Report 81-13, University of Massachusetts, May 1981.