International Journal of Computer Vision, 2, 209-250 (1989)

The Schema System

BRUCE A. DRAPER, ROBERT T. COLLINS, JOHN BROLIO, ALLEN R. HANSON,
AND EDWARD M. RISEMAN
University of Massachusetts, Amherst, Computer and Information Science

The *Schema System* embodies a knowledge-based approach to scene interpretation. Low-level routines are applied to extract image descriptors called tokens, and these tokens are further organized by intermediate-level routines into more abstract structures that can be associated with object instances. The thousands of tokens that are extracted from an image can be grouped in a combinatorially explosive manner. Therefore, knowledge in the Schema System is not limited to the descriptions of objects; it includes information about how each object can be recognized. Object schemas control the invocation and execution of the low-level and intermediate-level routines with the goal of forming hypotheses about objects in the scene. The system described produces image interpretations based on two-dimensional reasoning, although nothing in the system organization and control strategies preclude the inclusion of three-dimensional information.

The schema framework exploits course-grained parallelism in a cooperative interpretation process. Schema instances run concurrently, and an object schema often has available a variety of strategies for identification, each one invoking knowledge sources to gather support for the presence of a hypothesized object. Interschema communication is carried out asynchronously through a global blackboard. In this way schema instances cooperate to identify and locate the significant objects present in the scene.

This paper first discusses the design of the Schema System with regard to the issues mentioned above, and then describes in some detail how that design is put into practice. The system uses the operators and algorithms of the VISIONS system for knowledge sources, and complex strategies for controlling the low- and intermediate-level KSs have been implemented. The ISR, an intermediate token database tuned for associative and spatial queries, is used for storing and manipulating image data. The result is an integrated, knowledge-directed system composed of modular knowledge structures that produces a two-dimensional interpretation of a digitized image. Interpretations of seven images from two natural domains are presented.

1 Vision and Knowledge Engineering

1.1 Introduction

Special-purpose vision systems have shown considerable success within their limited task domains (e.g., [30,56,39,42,3]). To date, however, there have been no general purpose systems that work effectively across a variety of domains. Why do special-purpose systems succeed where general systems fail? We believe that special-purpose systems have succeeded because they are better able to define, structure, and apply knowledge that is relevant to their task domain. Knowledge in vision includes domain-independent knowledge about occlusion, perspective, physical support, etc., as well as domain-dependent object knowledge about attributes and relations, and object-specific control knowledge for recognizing objects in scenes. Object knowledge can encompass three-dimensional structure, two-dimensional appearance, and geometric and co-occurrence relationships with other objects and object parts. Control knowledge addresses the efficient extraction, organization, and matching of image information to stored models and the ordering of constraints to insure efficiency and consistency in the evolving interpretation.

Systems working in restricted domains can bring very specific recognition and control knowledge to bear on their task. Very small domains admit the possibility of a complete world model, offering constraints that permit sophisticated inferences with comparatively little computation. General-purpose systems, on the other hand, require generalized knowledge and inference machinery that may not be applicable to a specific task with the same efficiency and reliability. Indeed, general-purpose techniques are often unable to solve nontrivial problems [13]. "Such problems are only solved by the use of a large, domain-specific knowledge base. It has become almost an axiom of artificial intelligence that powerful problem solving in any realistic domain requires a large amount of knowledge" [13]. This has led to the notion of "expert" or *knowledge-based* systems.

However, systems tailored for small domains require only a few types of description to recognize their restricted set of objects. As a result of their small vocabulary, such systems can afford to generate many tokens of each type. A more general system allowing less constrained contexts and multiple viewpoints must make fewer assumptions about what types of image descriptions may be necessary for later object verification. The computational issues associated with large-scale knowledge bases and a large search space of possible interpretations is particularly onerous for image understanding, already one of the most computationally intractable domains of artificial intelligence research. Processing typical color images (RGB) in real time, even at low resolution (256x256) and standard frame rate (30 frames per second), would involve processing over 5.8 million bytes of data per second. At the symbolic level, moderately sensitive line and region segmentation algorithms can easily produce over 4000 lines and 300 regions per image on typical natural scenes. Further computation of line and region features results in several tens of thousands of attribute values per image before the high or intermediate-level interpretation has ever begun.

Tsotsos [58] has applied a "complexity level" analysis to the computational requirements of general vision systems. Not surprisingly, spatial parallelism alone is insufficient. His analysis suggests that object models be hierarchial, and that object knowledge can be used to spatially constrain the search in the image, and to limit the number of image abstractions (tokens and features) computed. In addition he showed that a visual processing architecture could satisfy the timing constraints for human visual performance if object knowledge is used to constrain the processing.

Our primary design philosophy is that both knowledge and computation should be partitioned at a course-grained semantic level. Each schema is specialized to identify one particular class of object. A schema instance is invoked for each object instance hypothesized to be in the scene. These schema instances run independent concurrent processes, communicating asynchronously through a global blackboard when necessary. Depending on the hardware, these processes can be distributed among available processors. Each schema instance directs the application of general-purpose knowledge sources to gather support for its object hypothesis. The goal is to foster cooperation and competition among schema instances, resulting in a set of object hypotheses that are both semantically and spatially compatible.

The Schema System represents an attempt to build a more general-purpose vision system out of many special-purpose ones. It is a knowledge-based, high-level component of the VISIONS [26] image understanding system. The system goal is to interpret static, color images by identifying and locating the significant objects in the scene and identifying relevant object relationships. Knowledge bases currently exist for two natural scene domains: house scenes and road scenes.

1.2 Related Work

Other researchers have addressed the task of knowledge-directed vision (see the survey papers by Binford [6] and Tsotsos [57]). Nagao [42] used a variant of a production system in which the rules were actually complex visual subsystems. He was arguably the first to achieve a significant measure of success in a fairly complex natural scene domain. Ohta [44] used a more traditional production system, supplemented with certain nonmodular mechanisms for control knowledge. McKeown's SPAM [39] has a production-system knowledge base of over 500 rules, organized into five processing phases, for interpreting aerial images of airports. Glicksman [24] used a frame-based knowledge representation in his human-aided interpretation system, and Hwang [29] controlled a frame system by means of a blackboard-based scheduler. Researchers at Carnegie-Mellon University have developed a system for real-time navigation whose global organization is similar to the one described here to the extent that they use a set of continuous and concurrent visual "knowledge sources" which, like our schemas, communicate through a global blackboard [52,55].

At the University of Massachusetts, knowledge-based vision research has been pursued for more than a decade. The architecture of the VISIONS environment [26] , and its accompanying hardware expression in the Image Understanding Architecture [60], is organized by levels of abstraction, from low-level (or early) processing to high-level (or cognitive) processing. The experimental approach used in the VISIONS environment is to attack the data volume problem by abstracting the data at each level of representation and to attack the model proliferation and interaction problem by organizing the models and their recognition routines into discrete hierarchical subsystems. Early, bottom-up processing builds abstract symbolic descriptions of the underlying scene. Measured features of these descriptions cue initial object hypotheses, thereby invoking object-specific recognition mechanisms. Further processing of the image is performed in a top-down hypothesize-and-test fashion.

Hanson and Riseman [25] describe a frame-like "schema system" for computer vision. Their ideas were influenced in part by the earlier work of Arbib [2] and Minsky [40,41].[1] Later, Weymouth [63] demonstrated initial success at interpreting natural images with schemas running in a simulated distributed environment. The current UMass Schema System represents the continuing evolution of these ideas. Concerns about knowledge-base development time have led to increased standardization of schemas and the adoption of the blackboard as the only communication mechanism. At the same time, issues in distributed systems have led to splitting the blackboard into one global and many local blackboards.

1.3 Issues and Overview

A knowledge-based approach to image interpretation raises many difficult problems. We focus on two here (see [50] for a discussion of others):

- choosing a knowledge representation, and
- how to effectively utilize multiple processors.

The choice of a *knowledge representation* can greatly affect the ease with which a system is built, and its efficiency when running. Most knowledge-based systems encode information in the form of rules, frames, blackboard knowledge sources, or logic-based declarative languages. These representations are epistemologically equivalent; any system implemented in one could be rewritten using another. The differences are in efficiency, documentability, and ease of implementation/extension.

The main issues in *distributed processing* involve deciding how a given problem should be decomposed into components, and deciding how separate components should interact. Real-time, low-level visual processing probably cannot be realized without massively parallel architectures; however, coarse-grained parallelism seems appropriate for the high-level, semantic stages of image interpretation. Some issues in course-

[1] Our interpretation of the word *schema* is based primarily on Arbib's paper.

grained parallelism include deciding how to assign processes to processors, how to maintain consistency among processes (i.e., can more than one process be working on the same subproblem, and if so, how does the system choose between their possibly conflicting conclusions), and how process scheduling and resource allocation should be handled.

The next section presents the Schema System in more detail. Section 3 returns to the issues raised above, showing how the Schema System addresses them. Sections 4 and 5 describe a schema implementation, and show its interpretations of seven scenes. Section 6 presents our conclusions and future research directions.

2 The Schema System Design

2.1 Object Classes

The Schema System partitions both knowledge and computation in terms of natural object classes for a given domain. Part-of graphs of the object classes for the current road scene and house scene knowledge bases are shown in figures 10 and 12. Each class of objects and object parts has a corresponding schema which stores all object and control knowledge specific to that class. To identify an instance of the object class in an image, a *schema instance* is created.[2] A schema instance is an executable copy of the schema which runs as a separate process with its own state. The system's initial expectations about the world are represented by one or more "seed" schema instances which are active at the beginning of an interpretation (e.g., ROAD-SCENE, HOUSE-SCENE, or the more general OUTDOOR-SCENE). As these instances predict the existence of other objects, they invoke the associated schemas, which in turn may invoke still more schemas. From the instances thus spawned, the "successful" instances establish a set of objective hypotheses which constitute the interpretation of the image. At any point during the processing the current partial interpretation is made up of those hypotheses with the strongest measure of support from the data and from consistency relations with other hypotheses.

As the label OUTDOOR-SCENE suggests, object classes are not necessarily restricted to tangible objects; contextual or *scene* configurations also have schemas. A subcontext or "sub-scene" is like an object part; it is related to its parent scene or context in predictable ways, blurring the distinction between scene and object. At a sufficient distance, a house is an object to be recognized as a whole. At closer range, the same house also functions as a context for its parts (roof, wall, etc.). Instead of distinguishing the notions of object and context at the scheme level, the system gives contextual abstractions such as HOUSE-SCENE and OUTDOOR-SCENE their own schemas.

2.2 The Schema System

In the research reported here, the object representations and processing are primarily two-dimensional. There is an assumption that many objects are recognizable from standard viewpoints, and that 2D features of objects could be organized around these views [40]. Thus, from the point of view of the driver of a vehicle navigating down a two-lane road, certain tokens can be anticipated in the low- and intermediate-level token representation of an image. Given that the viewpoint is approximately assumed, tokens that support any portion of the following description can be made a part of the 2D schema model: (a) long, straight (or curved) diagonally converging lines bounding a road region that is centered in the lower portion of the image, (b) center-line markings that are different colored region(s) that are solid or divided into dashes and approximately bisect the road region, (c) vertically oriented signposts and telephone poles that touch the ground plane just to the side of the area representing the road, etc. Thus, the prototypical view provides 2D spatial constraints on the properties of line and region tokens extracted from the image.

One concern, of course, is that three-dimensional scenes and objects have an infinite number of viewing directions and, if the viewpoint is unknown a priori, then the approach could be un-

[2] The term *schema instance* will occasionally be truncated to *schema* where the context makes the reference clear.

tenable. In fact we do not believe that this is true. A related research effort [10,11] to precompile the 3D information in the possible views into a prediction for rapid indexing into the model base and 2D object recognition is underway. A *generic view* encompasses the range of viewpoints over which a set of tokens associated with object features will remain visible and for which a set of relations between these tokens will remain invariant. Thus, one generic view of the road scene might be formed from the range of viewpoints encompassing views from a few feet off the ground plane and within 20 degrees of the road axis down the road. This research has already progressed to the point where a prototype system for compiling the prediction hierarchy of a polyhedral model base from an unconstrained range of viewpoints exists and is being tested. Related research on generic views (or characteristic views) appears in [30,33,46,23].

The Schema System can be described from two perspectives: (1) the object-specific schemas and (2) the system level, describing how multiple schema instances interact. We will examine the system level in the following sections, and describe individual schemas in section 2.3.

2.2.1 Communication Requirements. Schema instances must be able to communicate to arrive at a consistent interpretation. While each schema can be viewed as a local expert subsystem for recognizing its associated object, this task often requires knowledge about other objects in the scene. A schema needs information about such things as the presence or absence of parts, contextually related objects, and possible occluding or shadowing objects as a necessary part of its own recognition process. Conflicting hypotheses which offer alternative explanations of image events are another important source of information.

It should be possible for any two schema instances to exchange information about a partial interpretation. However, it is impossible to predict in advance how certain objects will be related spatially within a 2D image. While spatial relations between structurally related objects can be predicted, coincidental relations such as occlusion or object label conflict can occur between any two object classes and will vary from one image to the next. We have concluded that a free flow of information between schemas is necessary so that new object classes and processing strategies can be easily included as the system evolves. At the same time, system modularity must be preserved. Knowledge engineering is an incremental process in which new schemas are added and old ones updated; changing one schema should not force related schemas to be altered. In the Schema System, information is exchanged through a global blackboard mechanism, while modularity is preserved by forcing public messages to conform to rigid, globally understood formats.

2.2.2 The Global Blackboard. The global system blackboard (see figure 1) allows schema instances to publish their contributions to the incrementally developing interpretation and to access the public contributions of other schemas. The Schema System benefits from the blackboard's decoupled communication in terms of flexibility and modularity. A schema can post messages without knowing who will read them; likewise, reading a message requires no information about when the message was or will be posted, or who posted it. A running schema can read a message left on the blackboard by a schema that is dormant or has ceased entirely. This means that a STOP-SIGN schema instance can use public information from a ROAD schema instance even though the later may have already finished processing, and that the STOP-SIGN schema could be added to the knowledge base with no modification to the ROAD schema even though the search for potential stop sign locations will depend on the location of the road in an image.

In the current system, the global blackboard is divided into sections, one section for each object class. Any schema can read from or write to any section of the blackboard; the division is for retrieval efficiency only. Dividing the blackboard into sections gives some assurance that a schema will not have to search through a large number of irrelevant messages. The sectioning of the blackboard by object class can be contrasted to the processing-level partitioning common in other blackboard and rule systems (e.g., [20,39]), a distinction that reflects the different semantic par-

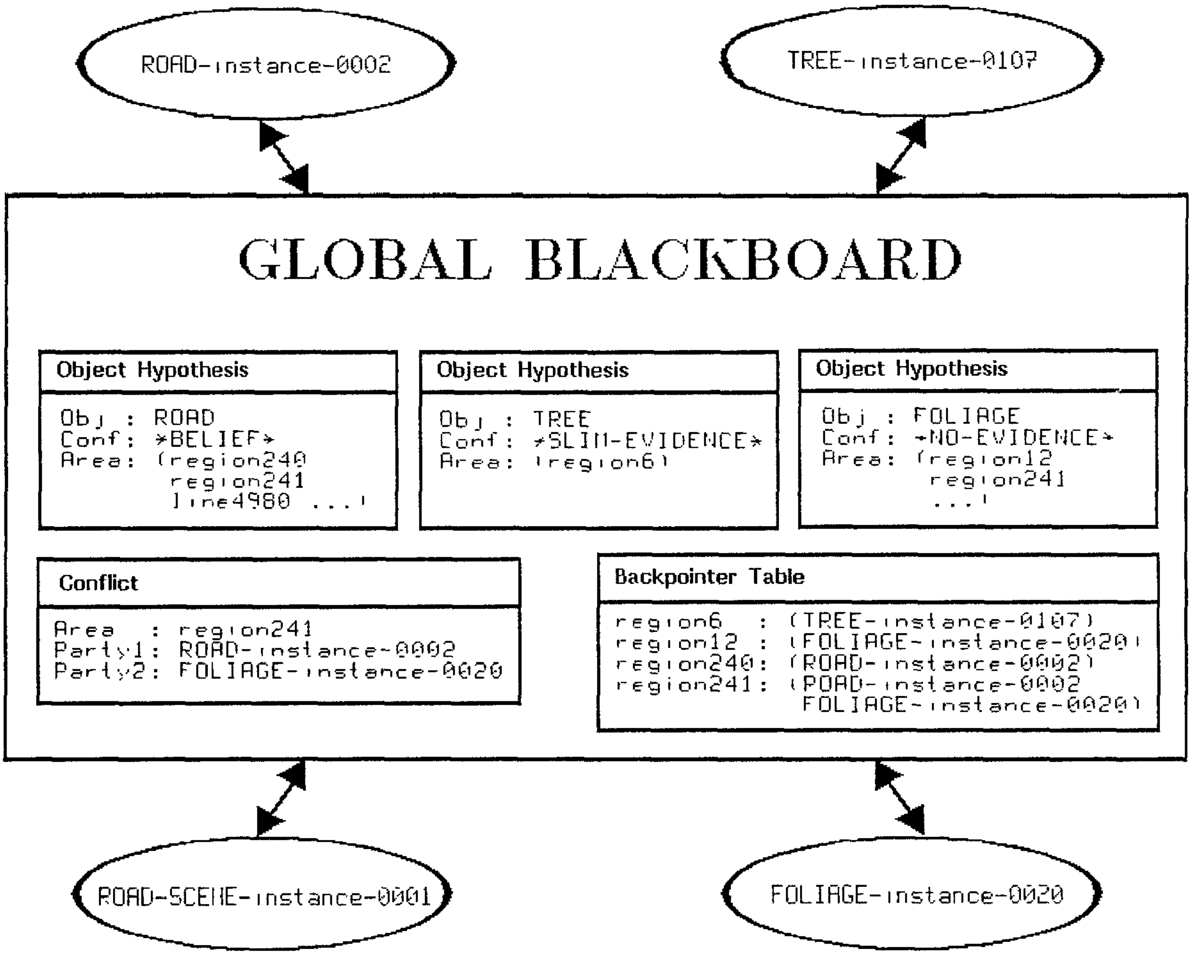


Fig. 1. The global system blackboard.

titionings of the two types of system. In a traditional blackboard system, the primary index of a knowledge source is its level of abstraction (although see [16] for a more sophisticated treatment); in the schema system, the primary index is the goal (object hypothesis) to which it contributes.

When a schema attempts to read a message that is not yet available, it may choose to suspend processing, or sleep, until an appropriate message is written. In general, a sleeping schema may wait for any number of potentially relevant events to happen. This is accomplished by writing predicates to the appropriate blackboard sections before suspension. A wakeup predicate should evaluate to true for any "interesting" messages, and cause reactivation of the schema. Whenever a message is written to a blackboard section, each of the predicates stored there is applied and those that evaluate to true have their appropriate schema instances awakened by the scheduler. This is similar to the use of *demons* in other AI systems, and avoids polling of the blackboard by sleeping schemas, at the cost of some slight overhead for message writing; the number of sleeping schema instances with predicates on any particular blackboard section is usually quite small.

2.2.3 Message Formats. All messages written to the global blackboard adhere to a small number of rigid formats which are consistent throughout the system. The most common message type is the *object hypotheses*, which includes

- the object class,
- a backpointer to the schema instance that posted this hypotheses,
- part/subpart links to other object hypotheses,
- a list of the portions of the image claimed or explained by this hypothesis, and
- a confidence value.

Schemas can use particular values of different fields of the object hypothesis structure to filter hypotheses when reading the blackboard. For instance, a schema can declare that it is only interested in those object hypotheses which pass a certain confidence threshold; hypotheses with lower confidence values will then be invisible to it.

Other message formats currently in use are (1) a spatial index from image area to the object hypotheses claiming it, and (2) a conflict message, posted whenever two incompatible hypotheses overlap spatially. Conflict detection and resolution in the Schema System is carried out in a distributed fashion. When a schema instance posts an object hypothesis, it also checks for conflicting hypotheses (by looking for spatial index messages associated with image areas that overlap its own hypothesis area). Overlapping object hypotheses of the same object class are ignored, as are hypotheses having a confidence level less than 'belief.'

When conflicting object hypotheses are found, a conflict message is sent to the schema instance that first posted the hypothesis. The two schemas then compete for control of the overlapped area. Where conflicts between two object types are predictable (e.g., tree trunk and telephone pole), recognition strategies designed to distinguish the two classes are used to resolve the conflict. Otherwise, the hypotheses' confidence values decide the issue, and if the confidences are equal then color and texture are used—a weak and inherently errorful method. Resolving conflicts between arbitrary hypotheses remains a difficult problem, particularly in systems with modular object knowledge.

2.3 Schemas

Although each schema is viewed as a special-purpose vision subsystem, building every schema from scratch would be an unmanageable task. The VISIONS environment provides a set of building blocks in the form of knowledge sources (see section 4.1). The ISR database (see section 4.2) provides representations of image events (e.g., region bitplanes) that can be matched to object descriptions (such as target graphs for the graph matcher; see section 4.1.3).

A schema contains the object-specific control knowledge. It is assembled by choosing the appropriate knowledge sources, and deciding when to apply them and how to interpret their results. More precisely, the schema designer (1) defines a *endorsement space*, that consists of the possible sources of positive and negative evidence for the presence of an instance of the object class; (2) provides one or more *strategies*, which are sets of paths and branches to traverse that space, taking into consideration the efficiency of the knowledge sources being applied and the likely quality and importance of the evidence returned; and (3) supplies a function to translate internal evidence from knowledge sources into a confidence value for the object's presence in the scene. Section 5.1 discusses the construction of schemas in greater detail.

2.3.1 Strategies. Strategies are simple control programs that run concurrently within each schema. They procedurally encode knowledge about which knowledge sources to apply and in what order. There are often several different methods of recognizing an object, some being more appropriate than others in certain situations. The HOUSE schema, for instance, might have one strategy for finding houses at a distance and another for recognizing them when nearby; certainly one could not use the same features to recognize houses from a distance of half a mile as from one hundred feet. Similarly, there could be one strategy for a frequently seen viewpoint and another, more computationally expensive strategy, for recognizing an object from any arbitrary position. Although not currently used, Burns discusses methods for automatically compiling view-dependent 2D features from 3D object descriptions ([10,11]; see also [30]). Such methods may in the future lead to the automatic construction of viewpoint-specific 2D recognition strategies from 3D object models.

Schemas can also contain strategies for recognition subtasks—e.g., one strategy for generating initial hypotheses and another for verifying them. Therefore, each individual strategy can be quite simple. One goal of our design is to constrain processing to the point where several schemas associated with different objects, and several strategies of each schema, may be executed in parallel.

One special strategy, associated with every schema instance, is the Object Hypothesis Maintenance strategy, or OHM. The OHM monitors the activity of the other strategies of the schema instance and updates the object hypothesis message on the global blackboard when necessary. In addition to its role as object hypothesis bookkeeper, the OHM also handles conflict detection and resolution related to the hypothesis.

2.3.2 Internal Hypotheses. Each schema instance develops and maintains *internal hypotheses* about possible instances of its object class in the scene. Internal hypotheses consist of (1) *tokens* representing image events; (2) a set of endorsements associated with the token; and (3) any other information, such as confidences, that might be useful. Tokens are abstract image descriptions representing image events. Low-level tokens are those extracted directly from the pixel data, such as regions or straight-line segments [5,9]; more abstract tokens result from grouping lower-level tokens into more complex entities [47,7], or from describing the relationship between tokens of different types [4,27].

In addition to tokens of image events, internal hypotheses contain a record of the evidence supporting or denying the presence of an object instance in the image. We currently use symbolic endorsements [15] as a medium for recording supporting evidence. The road schema, for instance, may invoke the IHS (see section 4.1) to determine if a region matches the expected color of road. The score returned by the IHS is then converted into one of three endorsements: *correct-road-color, neutral-road-color* (meaning ambiguous), and *wrong-road-color.* Similarly, an internal road hypothesis may acquire the endorsement *stop-sign-present* if a stop sign hypothesis with confidence of 'belief' is posted to the global blackboard. The set of possible endorsements spans the schema's endorsement space. The endorsement space is what the schema "reasons over" to determine how much an internal hypothesis is to be believed, and what knowledge source to run next. (See section 5 for more examples of endorsements and their use.)

To publish results on the global blackboard, the schema must first translate its internal hypothesis representation into the global object hypothesis format, which permits information to be exchanged without violating modularity. Translation is achieved using a schema-specific function (supplied by the schema designer) to map internal evidence into the global confidence scale. A schema maintains its internal representations of its hypotheses even after it has published them, so that if new positive or negative information is received, the schema can resume processing, either to recompute the hypothesis confidence due to new support, or to perform further analysis to strengthen a hypothesis weakened by loss of a source of supporting evidence. This provides a limited and local form of truth maintenance.

2.3.3 Representing Uncertainty. The representation and use of uncertainty is a subject that is receiving considerable attention and debate (see, for example, [14,15,45,51,62,64]). In our system, relevant information (or evidence) may be received from many sources, both bottom-up and top-down, and there is a need to reason across chains of inference involving uncertain object hypotheses. There are problems with Bayesian approaches in the propagation of probabilities that might involve closed chains of inference; in such cases it is possible for a probability associated with the initiating node that received the evidence to be updated based on an inferential chain from itself, and this is theoretically unsound. Alternatively, some researchers are attempting to apply the Dempster-Shafer formalism for evidential reasoning. While this has some attractive characteristics, such as an explicit representation of ignorance and the use of belief intervals in place of point probabilities, neither approach offers any solutions to the very difficult problem of lack of independence in the output of KSs. This is a serious issue for any approach that combines uncertain information from possibly dependent sources. In our case, the KSs are numerous, diverse, and complex and we have not sought to estimate the degree of dependence between subsets of KSs in order to better combine their outputs. This is a major research project in itself.

Basically, this paper does not deeply explore the theoretical issues associated with uncertainty. Confidence values are currently chosen to lie along a course, five-point ordinal scale ranging from 'no-evidence,' the lowest value, through 'slim-evidence,' 'partial support,' ''belief,' and finally 'strong-belief.' This five-point scale was chosen arbitrarily to represent a small number of degrees of belief to distinguish the different amounts and quality of evidence supporting a hypothesis. When combining evidence, we have used a heuristic mechanism that involves the specification of key endorsements (i.e., pieces of evidence) that are required to post an object hypothesis with a given confidence on the global blackboard. Subsets of secondary endorsements are used to raise or lower these confidences.

Of course, there is no guarantee that the system could not get caught in a loop of circular reasoning via a subset of schemas that post and withdraw hypotheses (or change associated confidence levels). However, one can put in mechanisms to detect such situations [18]. Our system is focused on the collection of additional endorsements in an attempt to avoid such situations. Thus, an object schema would be woken by a change in endorsement set or associated confidence level, and any remaining paths for evidence accumulation would be applied. Interestingly, the problem of circular reasoning has not occurred in our experiments.

2.3.4 Local Blackboards. A schema instance may contain many internal hypotheses, each of which must be available to all of its active strategies. At the same time, we have chosen not to allow unreliable and unverified hypotheses to be visible to other schema instances, in order to avoid propagating weak information and generating unnecessary processing. Therefore, each schema instance contains its own *local* blackboard, depicted in figure 2. Local blackboards are also partitioned into sections, where each section often corresponds to a different level of internal hypothesis abstraction. The level of abstraction of a hypothesis is determined by the abstraction level of its included tokens. For example, roadline hypotheses initially contain straight-line segments extracted from the image via a straight-line extraction algorithm.[3] Later processing builds more abstract tokens and hypotheses representing parallel pairs of these line segments, and after that regions are formed corresponding to the area between line pairs. Future development of the roadline schema might include the construction of 3D surface tokens from monocular, motion, stereo, or active range data.

The local blackboard is accessible to all the strategies making up a schema instance, but only those strategies. As a result, while messages to the global blackboard need to conform to a strict protocol, local blackboard messages can be highly schema specific, since the privacy of the schema's internal state is assured. This prevents the requirements of a large system from imposing undesirable restrictions on its subsystems. Each schema can maintain information in the manner most appropriate to its object. This permits the schema designer the freedom of any appropriate knowledge representation and control style, while at the same time protecting the Schema System from a plethora of public message formats.

2.4 Knowledge Sources

The Schema System interfaces with the rest of the VISIONS environment through knowledge sour-

[3] We refer to the white or yellow painted lines that mark road or lane boundaries as *roadlines*.

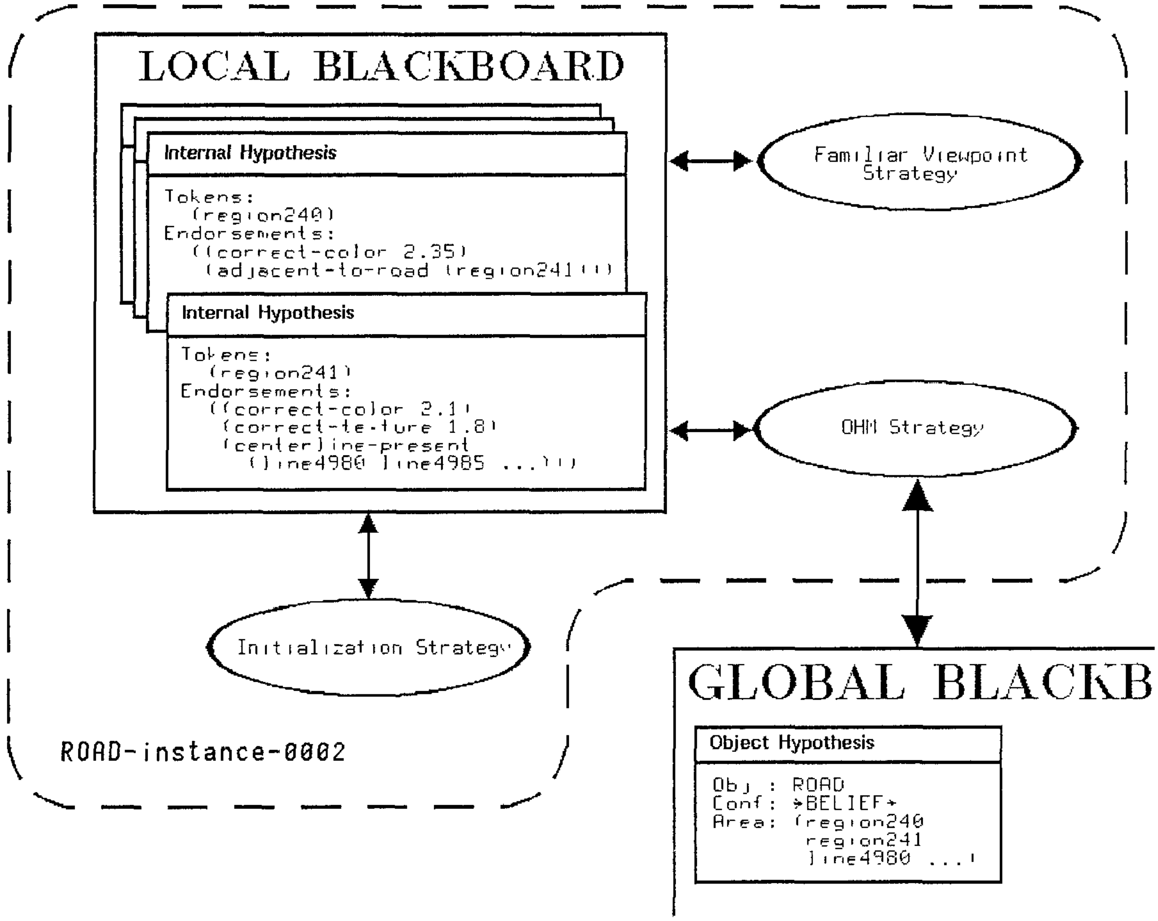


Fig. 2. A schema instance's local blackboard.

ces (KSs). In our system, knowledge sources are general-purpose programs or tools called by the schema strategies. While they may contain knowledge associated with their particular task, they do not contain object-specific knowledge, and thus are useful over a variety of objects and domains. Sometimes a KS is a complex subsystem that is itself a topic of research, as in the case of the *Constraint-Based Graph Matcher* or the *Initial Hypothesis System* (see section 4). It is important that the amount of computational work a KS performs be controlled by the calling strategy. For complex KSs this means that the major control factors must be parameterized, thereby allowing control decisions to be encoded in the schema strategy.

3 System Design Review

In this section we return to the issues discussed in section 1. In particular, we consider how the Schema System's design addresses the issues of knowledge representation, particularly control knowledge, and distributed processing.

3.1 Knowledge Representation

Knowledge in AI systems has typically been represented in frames, rules, blackboard knowledge sources, or logic-based declarative languages.[4] In

[4] For the sake of the arguments here, we will view logic-based

this section we discuss the relationship between the representation of knowledge in the Schema System and these alternatives, and argue that the Schema System architecture can be viewed as the natural evolution of a blackboard architecture in a distributed environment. Familiarity with these representational tools is assumed; background material may be found in [53] and [21].

3.1.1 Frames. Frames and *frame systems* have been popular in AI. As data structures, frames offer the benefits of record structures, slot-access procedures (demons or active values), and value inheritance. The representation features of frames are supported within the VISIONS environment: The Intermediate Symbolic Representation (ISR) data-base system supports record structure and slot-access procedures as well as associative access and indexing capabilities rarely found in frame systems, while in the Schema System, object hypotheses form a network of frames.

In a pure *frame system*, however, the frame is a control mechanism as well as a data representation. The control paradigm of a frame interpreter is essentially two-phase; a structure-matching phase and a forward-chaining phase. In the structure-marching or slot-filling phase, a candidate frame instance is chosen by some focus-of-attention criterion, such as "nearest to completion," and the data-base is searched in an attempt to fill the remaining slots in that frame instance. Where there are constraints on the slots that reference other slot values or other frame instances in the system, structure-matching becomes a form of subgraph isomorphism, with the accompanying potential for combinatorial explosion. Great care must be taken in defining constraints for a frame so that the search and match can be effective.

The forward-chaining phase is initiated by the attached *if added* demon when a slot is filled. It is more efficient that blind forward-chaining schemes in that a demon is not added to a slot until its action has been deemed meaningful, thus eliminating certain useless inferences. In addition, the representation of the problem domain in a frame system appears reasonably clean, since much of the procedural complexity is hidden in the attachments. This appearance can be extremely deceptive, however, when the attachments carry much of the burden of interpretation, precisely because the procedural behavior is concealed and not temporally coordinated. In the monolithic frame system, the designer possesses a snapshot of the network of data relations, but ordering, branching, and looping behaviors are extremely difficult to specify. This makes frame representation ideal for small, tightly integrated parts of a domain, where much of the knowledge can be captured in a restricted and relatively homogeneous implementation. However, imposing a frame representation blindly on *all* aspects of a computational domain can result in extreme inefficiency and opacity.

3.1.2 Rules. In *rule systems*, knowledge is encapsulated in if-then rules, or condition–action pairs, which interact with the body of data in working memory. Because of the large number of rules needed in a real-world task domain, additional control has been found indispensable and has usually been added by structuring the rule base into classes [54,44] or phases [39], or by implementing additional rule sets which make control decisions. This last approach defeats modularity and makes system modification difficult. Rule classes or phases represent this control explicitly, although the control decision to test a rule still resides in at least two separate locations (the rule's class declaration and any rules that invoke the class) so that a change to either location may have unforeseen effects on the computation. In addition, there is a trade-off between class size and the number of classes. At one extreme is a system with a few clearly marked classes or phases, each of which contain a large number of fine-grained rules (e.g., [39]); at the other extreme would be a large number of classes, each having a few complex, course-grained rules (e.g., [42]). This latter solution naturally leads to a blackboard

approaches as being equivalent to backward-chaining rule systems.

style of control, where *events* on the blackboard signal when a rule's condition part should be matched against working memory.

3.1.3 Blackboard Knowledge Sources. A *blackboard knowledge source* (BBKS) is the equivalent of a rule[5]. In addition to the condition and action components, each BBKS also declares a set of triggering blackboard events, one or more of which must occur before the BBKS's condition predicate can be tested. The declaration of triggering events gives the BBKS a measure of the state of the system which is more fine-grained than classes or phases, but more economical than exhaustively testing each rule predicate.

We have adapted much of the structure of the basic blackboard system, with a few significant adjustments. A great deal of work on control and scheduling in blackboard systems [28], [31] uses centralized control, so that a BBKS must have some way of indicating to the central controller why it is running, what resources it will consume, and what it might produce in the way of output. There may be a large number of parameters that a BBKS must "tweak" in an effort to influence the scheduler [28]. Where a group of BBKSs form a natural processing sequence or tree, it may become quite difficult for the system user to manipulate the scheduler in order to achieve the necessary effect. Furthermore, it is possible that a large investment in centralized control might prove fatal in a distributed system.

The lack of intra-BBKS continuity is perhaps the most annoying problem for the AI researcher. Large-scale perceptual systems do not always break down comfortably into chunks the size of the "READ RUN POST DIE" cycle of the BBKS instantiation. This problem was recognized as early as Hearsay-II, where terminating knowledge sources were allowed to dump their internal state onto the blackboard so that they could be resurrected and resumed at a later time [20].

3.1.4 Schemas. The Schema System gives a schema instance more intelligence, more autonomy and more continuity than a BBKS, thereby reducing the centralized control duties of the blackboard and the scheduler. The state of the interpretation is on the blackboard; if the available processing resources can be published in a similar manner, then there is no intrinsic reason why a schema cannot determine its own priority. The benefits we expect are greater efficiency in a distributed system (due to the reduction in control communication overhead and the elimination of a potential bottleneck) and reduced burden on the schema designer. Furthermore, schemas can read the blackboard as often as necessary and suspend processing for long or short periods with no difficulty. The schema designer can think of a schema simply as a program with a set of concurrent interpretation strategies, an internal measure of success, and a means of translating its internal results into a uniform public hypothesis. It should be noted that other vision researchers using blackboards have also seen the need for continuous concurrent processes that maintain their own state [52].

3.2 A Distributable System

The Schema System has been designed to run in a parallel environment. Schemas running concurrently allow as many hypotheses to be pursued simultaneously as available processors permit. Explicit parallelism is further expressed by strategies that may run concurrently within each schema. Since this system was conceived as a distributed system from the beginning, we have gone to some lengths to avoid certain bottlenecks peculiar to serial implementations of blackboard systems.

3.2.1 Distributed Control. The potential control bottleneck embodied in the centralized blackboard agenda and scheduler has already been discussed. As we pointed out, we hope to reduce the control bottleneck by distributing scheduling decisions among the schemas themselves. The issue of when the cost of control processing outweighs the benefits is still an open research question. In general, that process should run which in-

[5] Blackboard system knowledge sources (BBKSs) fall somewhere between our schemas and our knowledge sources (KSs) in both power and grain size. The two types of knowledge source (KS and BBKS) discussed here should not be confused. Our knowledge source is a function or set of functions that does not conceptually interact with the blackboard at all.

curs the least computational cost while providing the greatest contribution. Although cost is not too difficult to calculate, a potential contribution can only be assessed in the light of the goals of the computation.

In current blackboard systems [28], the solution is for the control knowledge sources to test, at one time or another, all of the other scheduled BBKSs in the system to determine which ones should be run in which order. Our position is that it will prove more economical, more modular, and more effective to publish the goal information that the control KSs in a blackboard would use, and let the schemas themselves decide whether or not to run. We believe that control in the Schema System should have a cost/benefit behavior no worse than that of the traditional blackboard, and generally should be better. Schemas suspend processing while waiting for a particular piece of control information, so that the processing queue (agenda) should be significantly smaller. Thus, control processing should benefit from the same modularity and opportunistic focus behavior as does the domain processing.

The Schema System is not yet large enough to begin testing our control assumptions. The current system goal is to continue the interpretation until all reasonable hypotheses have been examined. The system stops when it has interpreted everything it possibly can and has marked as unknown anything in the image it can not explain. Because of the localization of control, the system is able to use any generic process-scheduling mechanism. The current implementation of the Schema System relies on the TI Explorer™ process scheduler.

3.2.2 Distributed Communication. We have attempted to reduce the blackboard *information bottleneck* in two ways: (1) the small set of strategies within a given schema instance share a local blackboard; (2) the global blackboard is sectioned by object with efficient parallelism in mind. A schema's local blackboard can be located at a node where the schema's strategies will generally run. Since there is limited parallelism within a schema, there will be a small amount of potential processor-to processor communication (or memory contention in a shared memory system). Although in theory any schema can access any section of the global blackboard, in practice most schemas assess only a very few sections. Since the set of sections most likely to be accessed is known before a schema is run, it should be possible to determine a good distribution of the blackboard information and schemas across several processors.

In our experiment, ¾ of all blackboard messages were written to local blackboards rather than the global blackboard. This number is interesting, but preliminary; we have not yet begun to explore the space of factors that affect it. The local-to-global message ratio does seem to be influenced by the number of distinct objects present in the scene. This is to be expected, since fewer schemas mean less interschema communication. Other important factors include how much of the image was successfully interpreted, the complexity of the individual schemas in terms of knowledge sources, and the frequency with which a schema's global hypothesis is updated to reflect changes in the internal hypotheses. Regarding the last factor, the current method employed is to update the external hypothesis whenever its confidence level or token field is affected by a change to an internal hypothesis. This results in a large number of global blackboard messages, since each update involves erasing the old hypothesis and posting the new one.

3.2.3 Distributed Implementations. We currently simulate parallelism on a TI Explorer™/Lisp machine. A forthcoming implementation of the Schema System will work on a Sequent Balance™/21000 parallel processor. This 16-processor (expandable to 30) shared-memory machine will run a local implementation of Concurrent Common Lisp. Ultimately, the target machine of our system is the proposed Image Understanding Architecture (IUA) [60], which would provide 64 or more parallel symbolic processors with channels to more massively parallel environments for low- and intermediate-level vision algorithms. The IUA would offer an excellent environment for the Schema System since there would be enough processors to support a large number of schemas running concurrently.[6]

While there are many issues inherent in distributed processing, we expect that the system design will free us from some of the problems of adapting to a parallel environment. We expect the Schema System's distributed blackboard to reduce memory contention on any set of distributed processors with shared memory; we also expect that it will reduce, although not eliminate, the bottlenecks expected in message-passing multiprocessors.

4 Knowledge Sources

This section describes the system's knowledge sources. The VISIONS system [25,12] provides traditional image analysis techniques, such as region segmentation and straight-line extraction, that serve to construct the initial token representation of the image. More recent research efforts in such areas as perceptual organization and knowledge-based resegmentation also supply useful tools for the further construction of more abstract tokens [47,34]. All of the KSs, including the region segmentation and line-extraction routines, are parameterized so that they may be activated top-down from the schema control strategies.

The ISR data base, although not really a knowledge source, is also described in this section. It provides a set of common representations for the results of all KSs, as well as providing data storage. Many of the knowledge sources presented here rely on the ISR for its fast associative and spatial retrieval capabilities.

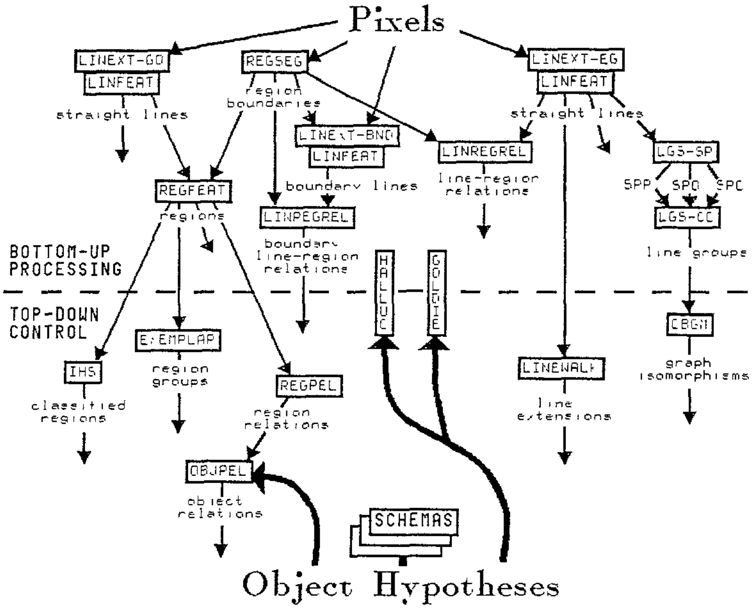
This section provides the information necessary to understand the system components and to compare the system to other knowledge-based image understanding efforts. A second goal is to emphasize to the reader the full complexity of the scene interpretation problem. Techniques exist for extracting tens of thousands of image events (lines, regions, curves, surfaces, etc.) from any given scene. These in turn can be grouped in a combinatorially explosive number of ways. In addition, the semantics of a given event are not fixed; a line may be the edge of an object, a reflectance discontinuity (i.e., surface marking), a shadow boundary, or simply a texture element, depending on the objects in the scene. The goal of the Schema System design is to handle these ambiguities by using object knowledge to provide top-down control of the vision process and to partition it for parallel execution.

4.1 Knowledge Sources

Knowledge sources are processes that generate the levels of abstract image descriptions required by an image understanding system. They can be grouped roughly by the level of abstraction at which tokens are generated. From the point of view of an interpretation system, even low-level processes such as segmentation algorithms can be viewed as KSs. Low-level KSs operate directly on pixel data to produce symbolic representations of the primitive structure of the scene. KSs at the intermediate level group symbolic descriptions and fuse information across modalities. The highest levels of abstraction are manipulated by the schemas themselves. A schema combines relevant KS results into support for its object hypothesis, invoking the appropriate KSs whenever those results are not already available. Figure 3 shows the flow of data through the system from the pixel arrays through increasingly more abstract levels of representation. Flow of control is not directly shown in this diagram. Above the dashed line, processing is currently data driven and the flow of control follows the flow of data; below the dashed line, knowledge sources are selectively invoked by the schemas.

Most knowledge sources can be used in either a data-driven or model-driven fashion. The low-level KSs are initially run in a data-driven manner. Their default parameters are set so that they are fairly robust over a wide range of image domains. Once the schemas have detected an image content, or have a particular goal to achieve, low-level KSs can be selectively re-run with their parameters set to a more discriminating level. Most intermediate-level KSs are model

[6] However, the initial prototype will only be a 1/64 slice of the machine with a single symbolic processor.



Abbrev	KS Title
EXEMPLAR	Exemplar Extension
CBGM	Constraint-Based Graph Matcher
GOLDIE	Goal-Directed Intermediate-Level Executive
HALLUC	Top-Down Token Creation
IHS	Initial Hypothesis System
LGS-SR	Line Grouping System for Binary Spatial Relations
LGS-CC	Line Grouping System for Connected Components
LINEWALK	Line Extension
LINFEAT	Straight Line Feature Extraction
LINREGREL	Line-Region Relation Routines
LINEXT-BND	Straight Boundary Line Extraction
LINEXT-EG	Straight Line Extraction by Edge Grouping
LINEXT-GO	Straight Line Extraction by Gradient Orientation
OBJREL	Object Relation Routines
REGFEAT	Region Feature Extraction
REGREL	Region Relation Routines
REGSEG	Region Segmentation by Localized Histograms

Fig. 3. Data Flow: From Pixels to Object Hypotheses. All data arrows not going into a knowledge source are examined directly by the schemas. The dashed line marks the boundary between bottom-up and top-down knowledge source control.

driven, controlled by the schema strategies that invoke them. Typically, they group symbolic tokens created by the low-level routines, building more abstract tokens as a result. The combinatorics resulting from the quantity of low-level tokens and the variety of relations that can be used to group them are such that top-down control is necessary. Schemas can constrain the combinatoric growth by restricting application of these KSs to subimages and structures that appear to exhibit the desired properties.

4.1.1 Low-Level Knowledge Sources. The current set of low-level processes operate on the numerical pixel arrays, or *image planes*, to segment the image into region and line tokens, and to calculate features for these tokens. These operators are all part of the larger VISIONS environment [25,12]. As we come to better understand how these low-level routines can be parameterized to take advantage of top-down expectations, these processes will become a more integral part of the knowledge-based processing [34,35]. Currently, the low-level processes include the following:

- *Region Segmentation by Localized Histograms (REGSEG)*— Derives a region segmentation using a localized histogram technique. Histogram cluster labels (defined by peaks and valleys) of regular subdivisions of each color

Table 1. Attributes computed for region tokens.

	Feature	Measures	Feature Description
	RAWRED	MEAN, SD	red (R) values for pixels in region
	RAWGREEN	MEAN, SD	green (G) values
	RAWBLUE	MEAN, SD	blue (B) values
	INTENSITY	MEAN, SD	intensity values $((R + G + B)/3)$
C	EXRED	MEAN, SD	excess red values $(2R - (G + B))$
O	EXGREEN	MEAN, SD	excess green values $(2G - (R + B))$
L	EXBLUE	MEAN, SD	excess blue values $(2B - (G + R))$
O	HUE	MEAN, COUNT	HSV basis hue, [22]
R	SATURATION	MEAN, SD	HSV saturation, [22]
	VALUE	MEAN, SD	HSV value, [22]
	TVY	MEAN, SD	YIQ basis $Y = W - Bk = (.30R + .59G + .11B)$
	TVI	MEAN, SD	YIQ $I = R - cyan = (.60R - .28G - .32B)$
	TVQ	MEAN, SD	YIQ $Q = magenta - G = (.21R - .52G + .31B)$
	HEDGE	MEAN, SD, COUNT	horizontal edge strength per unit area
	VEDGE	MEAN, SD, COUNT	vertical edge strength per unit area
	UPDEDGE	MEAN, SD, COUNT	upper diagonal edge strength per unit area
T	LOWDEDGE	MEAN, SD, COUNT	lower diagonal edge strength per unit area
E	SUMEDGE	MEAN, SD, COUNT	sum of directional edge strengths
X	ENERGY	MEAN, SD, COUNT	energy of the intensity histogram
T	ENTROPY	MEAN, SD, COUNT	entropy of the intensity histogram
U	LINE-DENSITY-1	COUNT ÷ AREA	lines of length [0,3] and contrast $\geqslant 10$
R	LINE-DENSITY-2	COUNT ÷ AREA	lines of length [0,5] and contrast $\geqslant 5$
E	LINE-DENSITY-3	COUNT ÷ AREA	lines of length [0,10] and contrast $\geqslant 3$
	LINE-DENSITY-4	COUNT ÷ AREA	lines of length [0,3] and contrast $\leqslant 3$
	LINE-DENSITY-5	COUNT ÷ AREA	lines of length [0,5] and contrast $\leqslant 5$
	LINE-DENSITY-6	COUNT ÷ AREA	lines of length [0,10] and contrast $\leqslant 10$
S	COMPACTNESS	RATIO	ratio of square of region perimeter to region area
H	MBR-ANGLE	ANGLE $[0,\pi/2]$	orientation of the minimum bounding rectangle
A	MBR-FILL	PERCENT	ratio of region area to area of the MBR
P	HEIGHT-TO-WIDTH-RATIO	RATIO	log of region height to width ratio
E	PERIMETER-RATIO	RATIO	ratio of region perimeter to MBR perimeter
	PIXEL-COUNT	COUNT	number of pixels in region (area)
S	REGION-PERIMETER	LENGTH	length of region perimeter
I	HEIGHT	LENGTH	height of region in pixels
Z	WIDTH	LENGTH	width of region in pixels
E	MBR-HEIGHT	LENGTH	height of the minimum bounding rectangle
	MBR-WIDTH	LENGTH	width of the minimum bounding rectangle
L	EXTENTS-MIN	ROW, COL	upper lefthand corner of region
O	EXTENTS-MAX	ROW, COL	lower righthand corner of region
C	CENTROID	ROW, COL	position of the region centroid

pixel plane are used to determine regions. The results are combined using a region intersection operation over all color planes. Constraint-based region merging is then applied to remove division boundaries and rejoin overfragmented regions [5,43,36].

- *Region Feature Extraction (REGFEAT)*—calculates features for a region from the color and intensity pixel planes, and the set of straight lines extracted by the LINEXT-GO KS described below. Computed features can be divided into five categories: color, texture, shape, size, and location (see table 1). The line tokens participate in the derivation of texture features—lines are divided into groups by similar length and contrast, and the number of lines from a group falling within the region is taken as a measure of texture.

- *Straight-Line Extraction by Gradient Orientation (LINEXT-GO)*—extracts a set of straight-line tokens from the image. Pixels in the intensity plane are labeled via coarsely quantized gradient orientation. A connected-components algorithm is then run to determine line-support regions (i.e., a set of pixels with an intensity surface that supports the presence of a straight line). Representative lines are extracted by intersecting a plane corresponding to the average intensity of a line-support region with a least-squares planar fit of the underlying intensity surface of that region [9].
- *Straight-Line Extraction by Edge Grouping (LINEXT-EG)*—extracts a set of straight-line tokens from the image extracting local edges and then hierarchically grouping them via geometric relations that were inspired by the Gestalt laws of perceptual organization [61,7]. The initial edges are the Zero crossing points of the Laplacian of the intensity plane; see LINEXT-GO for another possible set of initial edges. In an iterative process, two edges are linked and replaced by a single edge if their end points are close and their orientation and contrast are similar.
- *Straight Boundary-Line Extraction (LINEXT-BND)*—runs the LINEXT-GO KS described above with a restricted set of input pixels corresponding to those pixels lying along region boundaries. The resulting straight lines provide a description of the region boundaries.
- *Straight-Line Feature Extraction (LINFEAT)*—calculates features for a straight-line token given the line's end points.

4.1.2 Discussion of Low-Level Knowledge Sources. Figure 4 shows a typical road scene image. Segmentation results from the REGSEG and LINEXT-EG KSs appear in figures 5 and 6. The region segmentation is fairly good, since most of the significant object boundaries are present. Notice, however, the fragmentation of the road as it goes into the shadows, and the general fragmentation of the tree trunk and foliage areas. This is to be expected from a routine meant to group pixels of homogenous color and brightness without any

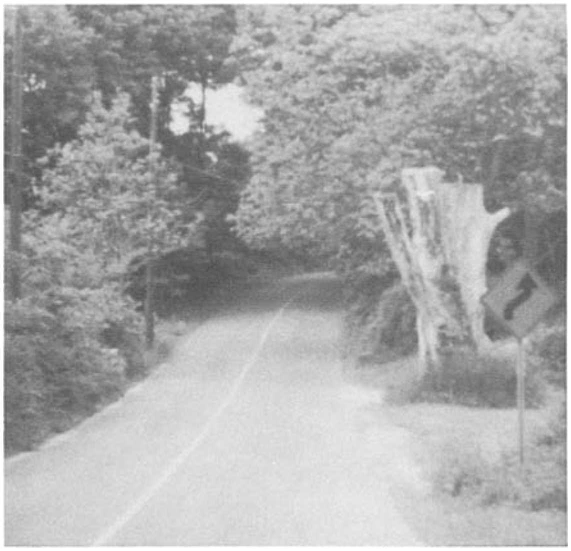


Fig. 4. A typical road scene image.

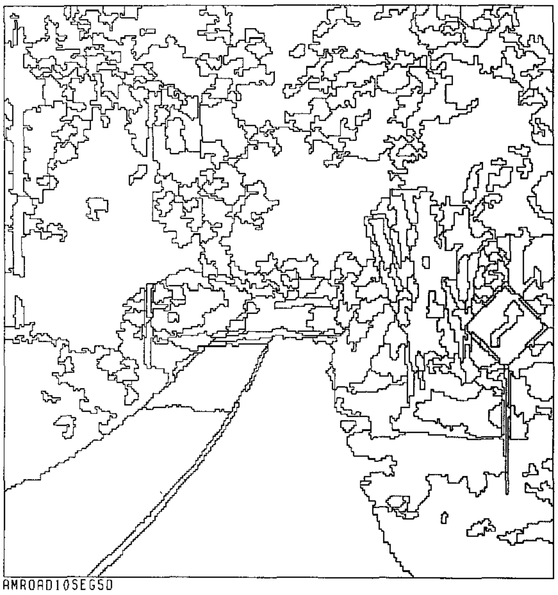


Fig. 5. REGSEG KS region segmentation of the image in figure 4.

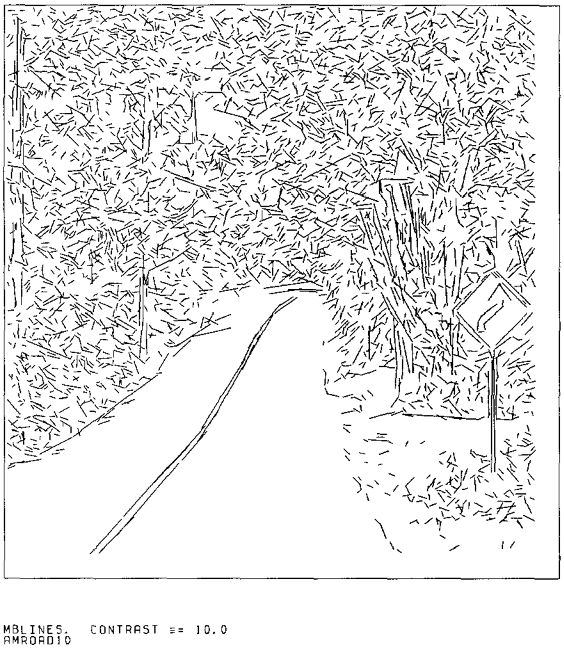


Fig. 6. LINEXT-EG KS straight line representation of the image in figure 4.

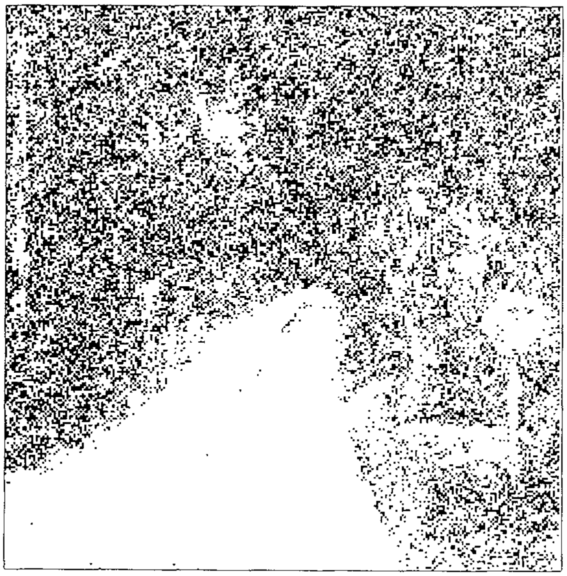


Fig. 7. Midpoints of lines extracted via LINEXT-GO and used for LINE-DENSITY-2 texture measure (length ⩽ 5, contrast ⩾ 5).

knowledge of object attributes and relations. The results from LINEXT-EG are highly fragmented, reflecting the textual variation and lack of straight-line structure in many outdoor scenes. Yet these tokens include strong parallel lines along the length of the road center-line, whereas REGSEG breaks the center-line into two pieces and loses it completely in the shadows. There is a tendency for long thin objects to be fragmented by the REGSEG KS, but picked out with more success by the LINEXT-EG KS. Knowledge such as this, about the behavior of knowledge sources, is needed in the interpretation process, and should be embedded in the schema control knowledge by the user.

Straight lines extracted by the LINEXT-GO KS are used to compute texture measures for the REGSEG region tokens. The LINE-DENSITY-2 texture measure (shown in figure 7; see table 1 for the definitions of the different texture measures) is the density of midpoints of LINEXT-GO lines having length ⩽ 5 pixels, and a contrast ⩾ 5 brightness levels.

All of the low-level KSs described here are run in a data-driven manner; that is they are run everywhere in the image with parameters set at default levels known to give usable results over a wide range of images. Table 2 summarizes some statistics collected for the descriptions these KSs produce on the road-scene and house-scene images for which results are presented in section 6. (The line-region intersection tokens listed in the table are computed by the intermediate-level LINREGREL KS described later). The data represented in the table is the current starting point for a Schema System interpretation, and should give the reader a feel for the amount of data involved.

4.1.3 Intermediate-Level Knowledge Sources. Intermediate-level knowledge sources operate on the results of earlier knowledge sources to build more complex representations of the data. Often, the presence or lack of a KS for extracting a particular image event has determined the programmer's success in writing a schema. We originally imagined a large library of special-purpose intermediate KSs. Instead, we have grouped the intermediate-level knowledge sources into five

Table 2. Size of the intermediate symbolic representation after bottom-up processing.

Image Name	Number of REGSEG Regions	Number of LINEXT-EG Lines short/other	Number of Line-region Intersections	Number of LINEXT-BND Boundary Lines short/other	Number of Boundary Line-region Intersections	Data Size Including Token Features (in Mbytes)
ROAD1	427	4019/1051	2736	3733/379	1415	2.4
ROAD2	311	4209/996	2111	3246/242	853	2.1
ROAD3	222	3744/733	1112	1305/187	775	1.5
ROAD4	356	4364/949	2354	3730/285	980	2.3
HOUSE1	305	2843/878	1793	1174/405	1438	1.5
HOUSE2	168	3522/900	1823	2239/242	711	1.6
HOUSE3	165	2675/845	1255	965/241	836	1.2

Notes: Short lines are those whose length is less than 5 pixels.
Line-region intersection features are not calculated for short lines.
All images have a resolution of 256 × 256 pixels.

(somewhat arbitrary) categories: feature-based classification, perceptual organization and grouping, geometric model matching, token relations, and knowledge-directed resegmentation. Some of these categories have become ongoing research efforts by some of our colleagues at UMass. In each category many independent functions have been replaced by a single, parameterized system. For example, parallel line pairs and collinear line pairs, which had been extracted by different routines, are now computed by a single parameterized subsystem. This simplifies the programming task, since there are fewer programs to interface with. Moreover, the parameters provide the schemas with a language in which to express desired image events; in the case of parallel lines, the schema can specify "how parallel" in a top-down manner.

The five categories mentioned above have been helpful in the experiments reported on here, but will not necessarily be sufficient for all domains. In the future we expect to add categories to help with 3D reasoning, motion parameters, and depth (via motion, stereo, or direct ranging). Currently available intermediate-level KSs include the following:

1. Feature-based Classification

- *Initial Hypothesis System (IHS)*—derives feature value constraints from statistical measures applied to hand-labeled region segmentations of a set of training images; the constraints are then applied to new region segmentations. For a given region and object, the degree of consistency between the region's feature values and the object's stored feature value constraints can be computed on demand. The IHS can be invoked by giving it a region, in which case it returns a set of rank ordered objects or by giving it an object, in which case it returns a set of rank-ordered regions [38,48,27].
- *Exemplar Extension (EXEMPLAR)*—The appearance of many objects, such as grass and sky, can be expected to vary less within an image than between images. The Exemplar KS takes exemplar regions from a specific image, and returns other regions from the same image which appear similar. The exemplar regions can be found by choosing the most reliable of the object-region associations returned by IHS, for example. To be precise, EXEMPLAR takes as input a list of region features which define an n-dimensional feature space, and a list of regions. Each region in the image can then be represented as a point within the feature space. The location of the initial exemplar in the feature space is obtained by averaging the normalized feature values of the given

regions. An iterative heuristic search technique which approximates hill-climbing is then used to locate the cluster in the feature space nearest to the initial exemplar, and all the tokens within the feature-space cluster are returned.

2. Perceptual Organization and Grouping
   - *Line-Grouping System (LGS)*—groups straight lines into spatially related pairs and connected sets. The LGS consists of two subsystems, the LGS-SR and the LGS-CC [47]. Note that the LINEXT-EG KS can also be used for line grouping.
     a. Line-Grouping System for Binary Spatial Relations (LGS-SR)—groups straight lines into spatially related pairs using constraints on relational measures. The output is a graph of the straight lines and their relations. Relations currently under investigation and in use are spatially proximate parallel (SPP), spatially proximate collinear (SPC), and spatially proximate orthogonal (SPO) [47].
     b. Line Grouping System for Connected Components (LGS-CC)—Applies a connected-components algorithm to the graph computed by LGS-SR, finding sets of lines such that a sequence of spatial relations holds between any two elements of the set. Certain global consistency constraints are maintained. For instance, when grouping collinear lines, every line in the set must have roughly the same orientation as every other line in the set, so that pairwise collinear connections will not be followed around a curve. The LGS-CC may currently be invoked on all seven nonempty combinations of the three binary spatial relations from the LGS-SR graph [47].
   - *Line Extension*—returns a set of lines that might be an extension of a given straight line, using LINEXT-EG. The search may be controlled by distance, relative angle, contrast, and direction parameters.
3. Constraint-Based Graph Matching
   - *Constraint-Based Graph Matcher (CBGM)*—Takes as input a *data graph* whose nodes are image tokens and whose arcs are tokens attributes and relations, and a *pattern graph* which describes a given object in terms of potential token attributes and relations [64,8]. The matcher verifies that the object's attributes and relations are present in the data. Feature constraints from the pattern are used to prune the data graph to a reasonable size. Since all monomorphisms from the model to the data graph are calculated, this KS is only invoked when both the pattern graph and the data graph (after pruning) are small. The number of possible matches in the worst case is roughly equal to $\Sigma_{i=1}^{n}$ *candidates(i)*, where *n* is the number of nodes in the pattern graph, and *candidates(i)* is the number of data objects whose attributes qualify them as candidates for model node *i*.
4. Token Relations
   - *Line-Region Relation Routines*—calculates region–line intersections, and measures features of the intersection such as the relative orientation of the line and region boundary, the percentage of the line that is interior to the region, the percentage of the line covered by region-boundary pixels, and the percentage of the region boundary covered by the line [4,49].
   - *Region Relation Routines*—calculates two-dimensional spatial relations between two region tokens. Examples are proximity relations such as adjacency, nearness, enclosure and overlap; and relative location relations such as above, below, left, and right.
   - *Object Relation Routines*—calculates relations between tokens associated hypotheses. Examples are object subpart/superpart relations and spatial object relations that are currently computed by using the REGREL library (see above) on object hypothesis region tokens. Future work will incorporate three-dimensional spatial relations.
5. Knowledge-Directed Resegmentation
   - *Goal-Directed Intermediate-Level Executive (GOLDIE)*—performs goal-directed region or line resegmentation on a subimage using parameter settings more appropriate to the given situation. This system is operational,

but not yet integrated with our system. GOLDIE is able to invoke any of the low-level segmentation routines, but in particular the REGSEG and LINEXT-EG KSs. Schema strategies will define the resegmentation goal using both object-specific knowledge (such as expected color or texture) and the context of the current partial interpretation (telling GOLDIE where in the image to concentrate) [35].

- *Top-Down Token Creation (HALLUC)*—used for strict "hallucination" of tokens, i.e., creation of tokens in a top-down manner without any direct reference to pixel values. Functions exist for creating a region that is the difference/intersection/union of existing regions; for creating a region from a set of points defining a closed polygon; and for creating a straight-line token from its two endpoints.

4.1.4 Discussion of Intermediate-Level of Knowledge Sources. Figure 8 shows some typical results from the IHS KS. In this figure, the IHS was invoked on the objects ROAD, ROADLINE, FOLIAGE, and TRUNK, and was asked to return all REGSEG regions having a positive correlation with the color and texture of the object, as determined through prior training. The regions returned are shown in black. The IHS uses constraints on the region token attributes to provide a quick guess as to which portions of the image warrant more expensive processing. An examination shows that the most obvious errors occur in the trunk hypotheses. If a large number of conflicts between tree trunk and other objects occur as a result, the trunk schema might reinvoke the IHS with stricter thresholds.

Figure 9 shows the result of a "bottom-up" invocation of the Line-Grouping System (LGS) KS on the straight-line tokens produced by the LINEXT-EG KS. In practice the LGS would be invoked to selectively compute these relations. Figure 9a shows all spatially proximate parallel-line pairs found by the LGS-SR KS; figure 9b shows the two largest connected component groups (in terms of number of lines) produced by the LGS-CC KS when allowed to group across all parallel, orthogonal, and collinear pair relations produced by the LGS-SR. The road-line schema uses the line extension KS while trying to find extensions of road-line hypotheses. Connected component graphs are used as input to the Constraint-Based Graph Matcher (CBGM) KS.

4.2 The Intermediate Symbolic Representation (ISR)

The ISR is a data base system tailored to the needs of intermediate-level computer vision. It appears to the user as a hierarchical frame system in which similar data events, such as all line tokens extracted from an image, can be stored as a group, called a *tokenset*, while still being accessed individually. This basic capability is then augmented for vision research in two ways:

- *Associative Access.* Tokens can be efficiently retrieved by their values as well as their name. For example, a knowledge source can access just the set of high-contrast lines in an image without having to check the contrast of each line separately. The set of tokens returned is called a *tokensubset.* This operation is optimized by organizing the feature values of a tokenset contiguously in memory, which preserves locality of reference. Intersection, union, and other functions over tokensubsets are also supported.
- *Spatial Data/Spatial Indexing.*— Most image events represent an abstraction of some portion of the image. The ISR supports a bit-map datatype which can be used to relate tokens to the pixels from which they are derived. It also supports operations over bit maps (e.g., intersection, union), as well as queries about them (e.g., "Get all tokens that overlap token X.")

5 The Schema Knowledge Base

When the schema designer develops a new-domain knowledge base, the first task is to identify which objects are likely to be seen in that domain and to specify their relationship. To this end, the construction of an object *part-of* graph can be helpful. Figures 10 and 12 are the *part-of* networks for the road scene and house scene

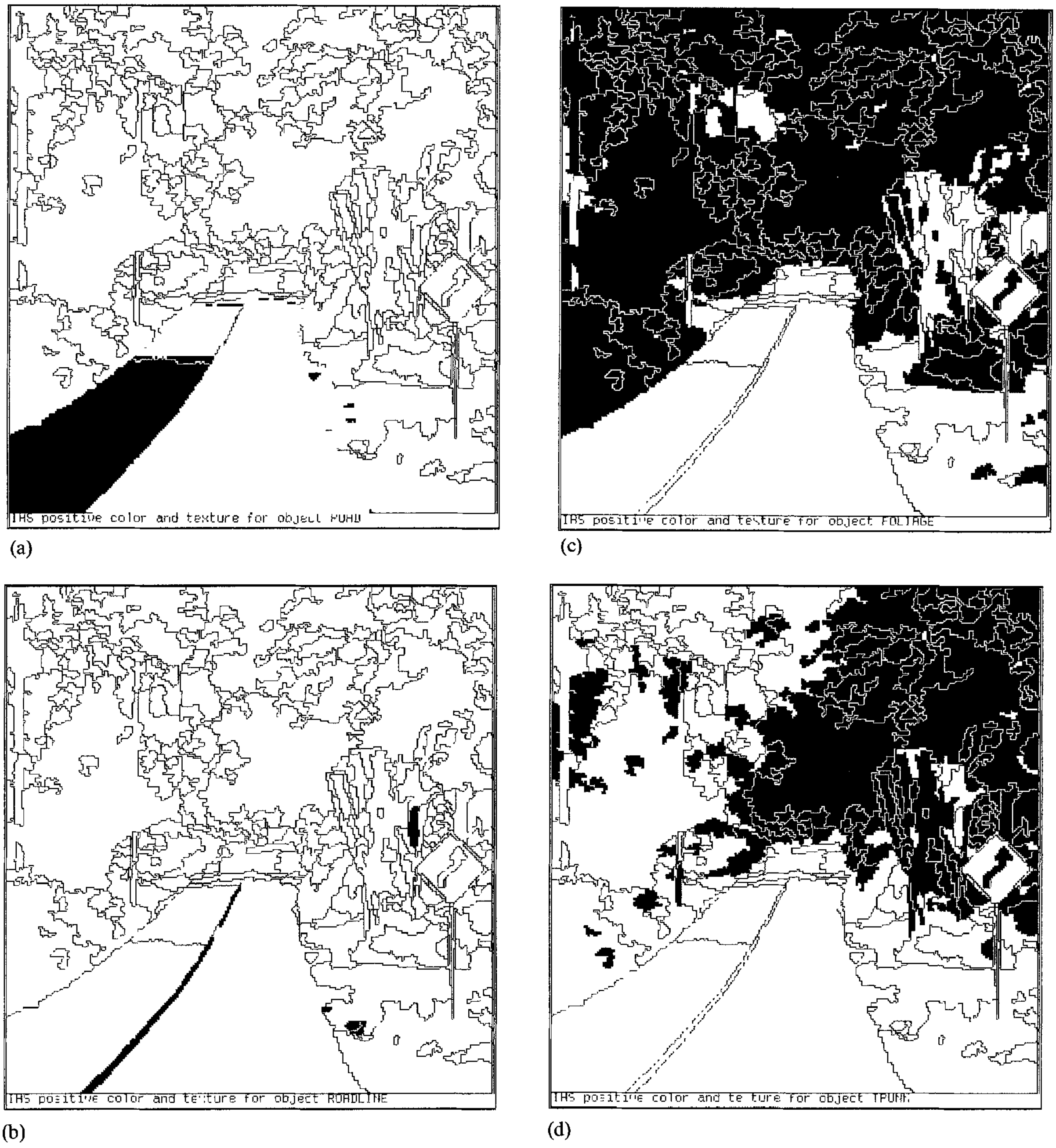

IHS positive color and texture for object ROAD
(a)
IHS positive color and texture for object FOLIAGE
(c)
IHS positive color and texture for object ROADLINE
(b)
IHS positive color and texture for object TRUNK
(d)

Fig. 8. Some results of the IHS KS when invoked on the regions from figure 5, looking for positive color and texture object scores. (a) ROAD regions. (b) ROADLINE regions. (c) FOLIAGE regions. (d) TRUNK regions.

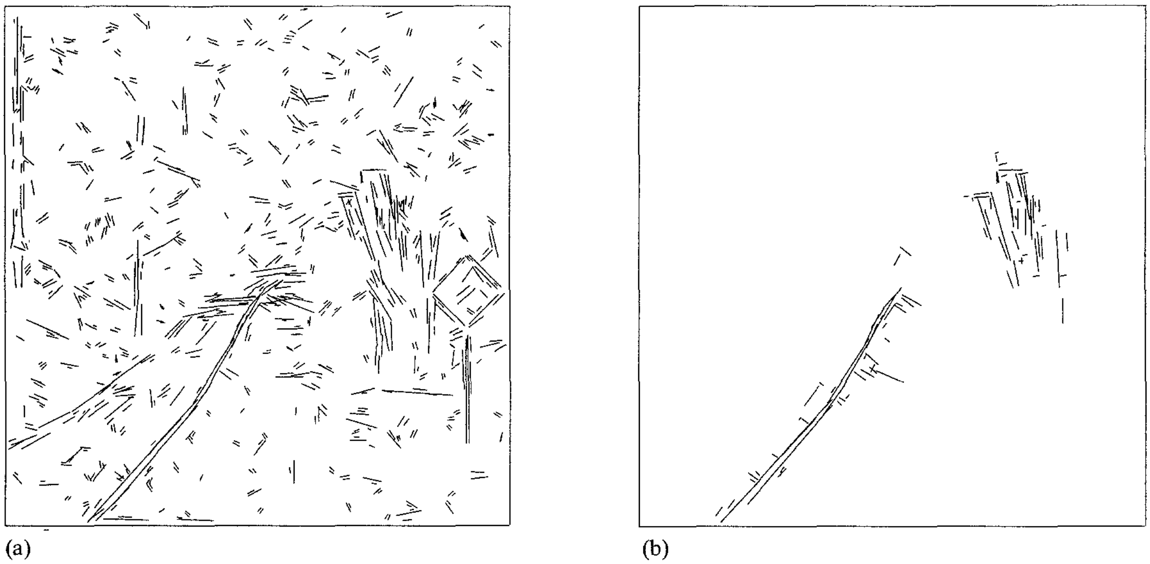


Fig. 9. Some results of the LGS KS when invoked on the straight lines from figure 6. (a) SPP pairs (see text). (b) Largest two con nected component groups from the COP graph.

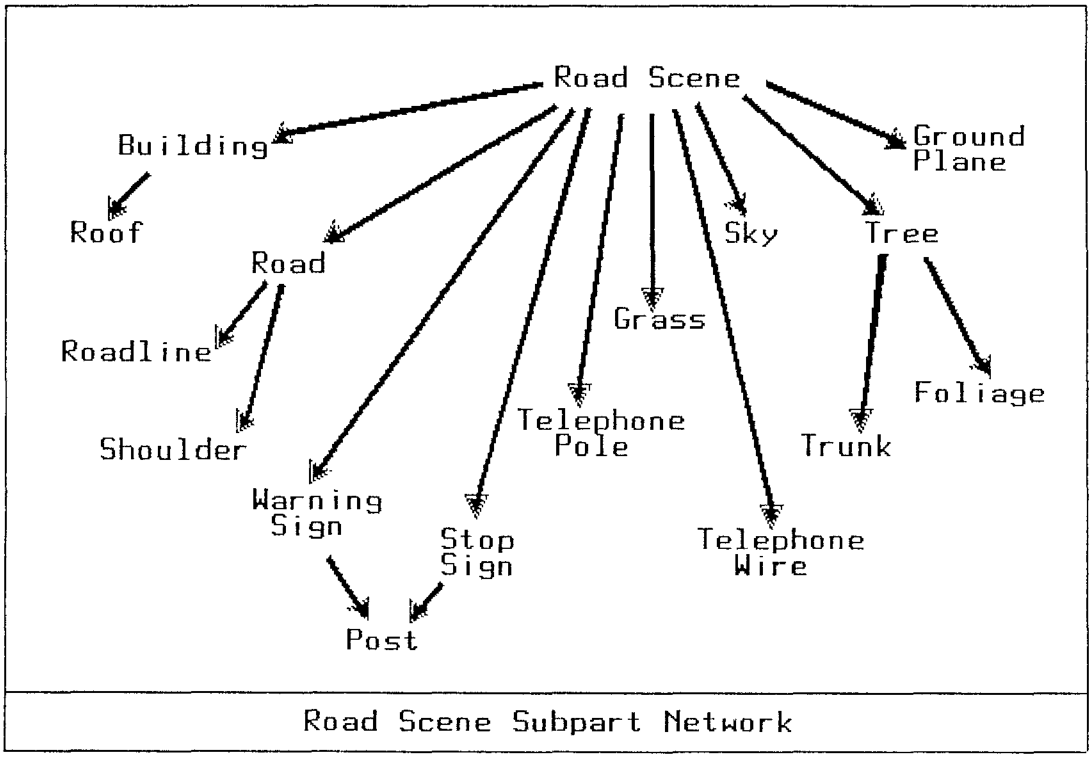


Fig. 10. Road Scene *Part-of* Network

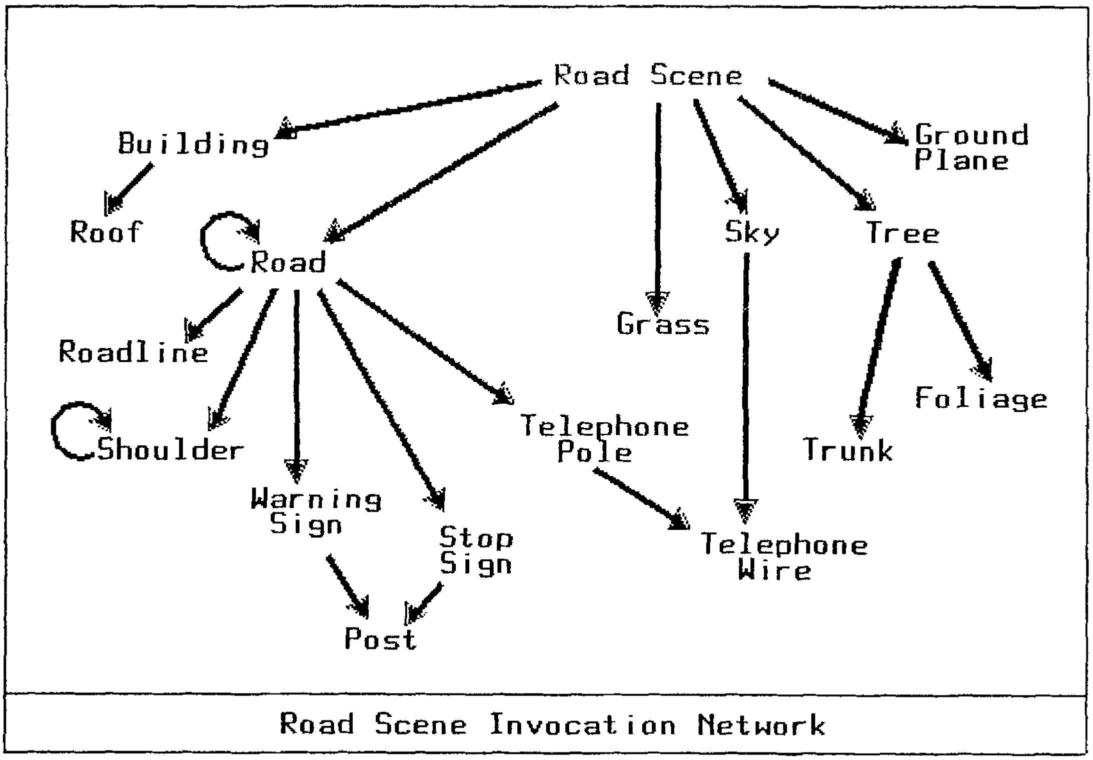


Fig. 11. Road Scene Invocation Network

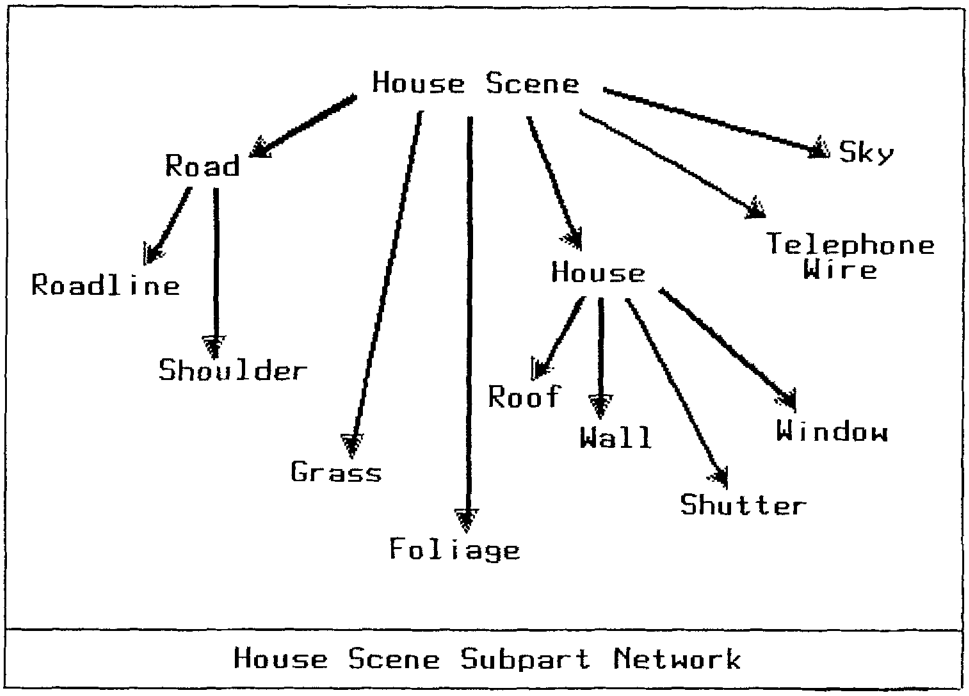
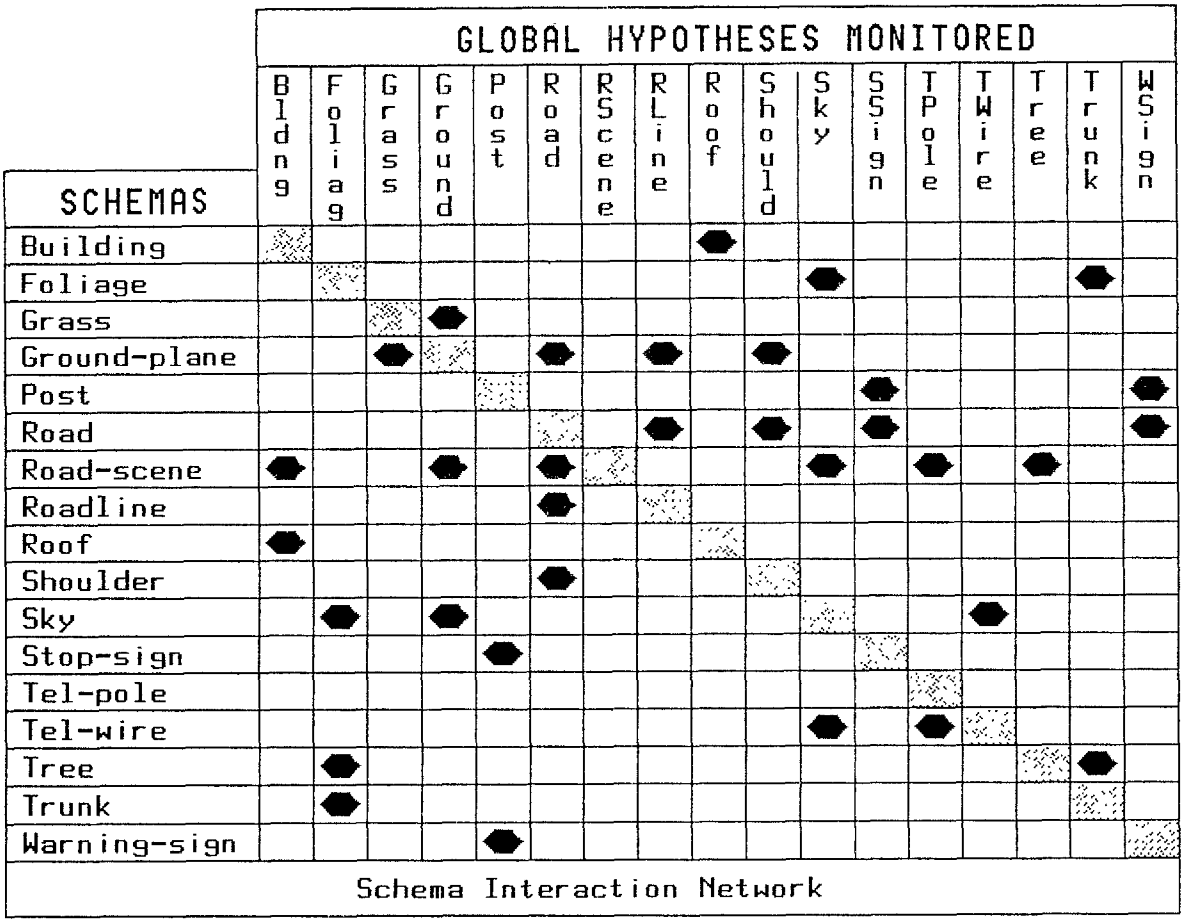


Fig. 12. House scene *part-of* network.

knowledge bases that are used to generate the results shown below. As was mentioned earlier, contextual scene objects are treated the same as physical objects, and appear in the *part-of* hierarchy.

A more faithful representation of the operation of the road scene knowledge base is shown by the invocation network of figure 11, depicting the possible schema invocation of other schemas. The invocation network indicates how goals are propagated through the system. In most cases the *part-of* network has been followed, since one important method of generating support for an object is to find one or more of its parts. Support is generated by invoking the proper object-part schema. In other cases, for instance when the sky schema invokes the telephone wire schema or when the road schema invokes one of the types of road sign schema, the *part-of* relationships have been abandoned. These invocation links express stronger contextual and spatial relations between objects than the corresponding *part-of* network links.

A third representation for viewing schema relationships is the interaction network, shown as an adjacency matrix in figure 13. This graph shows which object hypotheses are read by the various schemas as they are posted. The road schema, for instance, examines all roadline, shoulder, stop sign, and warning sign hypotheses, while its own hypotheses are read by ground plane, road scene, roadline, and shoulder schemas. Each schema can of course read the global hypotheses of any other schema, and will



	GLOBAL HYPOTHESES MONITORED																
SCHEMAS	Bldng	Foliag	Grass	Ground	Post	Road	RScene	RLine	Roof	Should	Sky	SSign	TPole	TWire	Tree	Trunk	WSign
Building	▨								●								
Foliage		▨									●					●	
Grass			▨	●													
Ground-plane			●	▨		●		●		●							
Post					▨							●					●
Road						▨		●		●		●					●
Road-scene	●			●		●	▨				●		●		●		
Roadline						●		▨									
Roof	●								▨								
Shoulder						●				▨							
Sky		●		●							▨			●			
Stop-sign					●							▨					
Tel-pole													▨				
Tel-wire											●		●	▨			
Tree		●													▨	●	
Trunk		●														▨	
Warning-sign					●												▨

Schema Interaction Network

Fig. 13. Road scene interaction network.

do so in the event that their hypotheses overlap spatially. Only the subset of interactions shown here, however, can provide positive support. In addition, not all of these interactions need to be satisfied. If an interpretation is run without either of the sign schemas, the road schema can still run without modification. The only difference would be that it could not derive support from the presence of road signs.

5.1 Schemas

The schema models the expected *appearance* of an object. This may be quite different from modeling the physical structure of the object. To take a pedagogical example, when viewing a car from the road at night, its shape and color are irrelevant—all the viewer will see are the headlights. The schema encodes information in terms of extractable image tokens and their attributes, and in particular the combinations of tokens and attributes that imply the presence of the object. This may lead to several distinct characterizations of the object, as its appearance changes with respect to different viewpoints and scales. There may also be multiple descriptions of a single scale and viewpoint, thus taking advantage of the redundancy inherent is visual processing. In the experiments shown here, assumed domain constraints have allowed us to limit each object to one viewpoint and scale.

The three elements of a schema—the endorsement space, the confidence function, and the control strategies—are used to form internal hypotheses on the local blackboard. Each characterization of the object is represented as a set of image tokens, token attributes, and token relations which, if satisfied, imply the existence of an instance of the object. The relevant attributes and relations are evidence for the object's existence, and are abstracted into endorsements, forming an endorsement space. The local blackboard is used to post internal hypotheses for token data with their associated endorsements. For example, for a telephone pole a pair of line tokens can be posted with the endorsement that they are parallel and vertical; then another KS can be used to determine whether the region between them is of low density (dark), and post that endorsement. Note that the token data can be posted without endorsements to allow sharing of data between strategies of a schema.

As a simplifying and normalizing approach across KSs, typically we have chosen to utilize three endorsements for each KS as evidence concerning a given attribute value or relation: one recording positive support, one negative support, and one a lack of information, as shown in the following examples. The schema designer is free, however, to use as many (or as few) endorsements as are necessary to capture the relevant distinctions. The goal of the schema is to satisfy one of these sets of endorsements, i.e., to give a token a set of endorsements that is mapped by the confidence function to one of the values "strong-belief" or 'belief.' Endorsements are acquired by invoking KSs that measure the attributes and relations in question. The power set of endorsements, although never represented explicitly, describes the abstraction space of possible evidence combinations over which hypotheses can be formed, and (as was previously mentioned) is referred to as the endorsement space of the schema.

The endorsement space characterizes the object and is the basis of the schema's confidence-mapping function. In general we have not assumed independence of the KSs when combining the evidence that the KSs provide. Each set of endorsements will be mapped onto the global confidence scale. However, the full enumeration of the endorsement space would be a burdensome task for the schema programmer. Thus, while the designer is free to choose any confidence-mapping function that is desired, we have used two methods for the experiments in this paper. The primary method has been to divide a schema's endorsements into a set of key endorsements and one or more sets of secondary endorsements. For each set of m endorsements, n endorsements ($0 \leqslant n \leqslant m$) will be specified in order to achieve a particular global confidence value (see table 3 for an example). In a few cases simple weighting (with a small number of possible weights) was employed to differentiate between stronger and weaker evidence when it was felt that the accumulation of independent evidence was

Table 3. The global confidence-mapping function for the roadline schema.

Label	Endorsement Subsets
A	:chain-pair-region-match
B	:location-match, :orientation-match
C	:correct-roadline-color, :acceptable-roadline-texture :bounded-by-lines, :near-road
D	:wrong-roadline-color, :wrong-roadline-orientation

Global Confidence Level	Required Endorsements
(strong-belief)	all of A & B, 3 of C, none of D
(belief)	all of A & B, 2 of C, none of D
(partially-supported)	all of B, 2 of C, none of D
(slim-evidence)	all of B, none of D
(no-evidence)	no endorsements or one of D

sufficient. The lowest global confidence value ‘no-evidence’ was reserved for internal hypotheses on the local BB whose previous endorsements had been removed. The methodology employed here was empirical in a test-and-refine approach, and we make no general claims for its adequacy.

Strategies control how the support space is searched. They allow the programmer to chain together sequences of KSs, and to abort such sequences when the KSs return unexpected or contradictory endorsements. In our experiments, a strategy contains a sequence of KSs whose invocation might provide an internal hypothesis with a set of endorsements that implies a confidence of “belief” of higher. At each step (i.e., KS invocation) the strategy has a set of restrictions, in the form of required and/or forbidden endorsements, that must be satisfied in order for the sequence to continue. Any strategies whose internal hypotheses can be simultaneously processed can be executed concurrently.

5.2 Example Schemas

Let us demonstrate the three schema elements with two examples from the road scene data base. The first example, *road shoulder,* is a simple schema, in part because regions are the only data representation used. The second, *roadline,* uses multiple internal data representations, and as a result is more complicated.

First we present the set of endorsements of the road shoulder schema. In New England, road shoulders (the ground surface adjacent to the road) are generally dirt or gravel, and lack a distinct structure or shape. Computable evidence for road shoulder is in the form of its spatial relationship to the road, and its color and texture measures. As a result, the support space for the road shoulder schema has six possible endorsements generated by invoking KSs: *correct-shoulder-color, neutral-shoulder-color, wrong-shoulder-color, correct-shoulder-texture, neutral-shoulder-texture,* and *wrong-shoulder texture.* The IHS, which compares the color and texture measures to the expected values for road shoulder, returns a degree of match: it returns a number between −10.0 (absolutely not a match) and 10.0 (a perfect match) for each category (color, texture). This range was qualitatively divided into three ranges for the road shoulder schema: a score above 1.0 would be considered support in that category, a score between −1.0 and 1.0 would be considered an ambiguous (or neutral) response, and anything lower would be counter-evidence.

The endorsement space also includes two endorsements, *near-believed-road* and *near-partial-support-road,* that are derived by reading the road schema’s hypothesis off the global blackboard. Global hypotheses are different from other sources of information, in that a schema strategy cannot “invoke” them. A strategy can invoke another schema, but it has no control over the hypothesis that schema will post, or when it will post it. Instead, when a strategy needs a global hypothesis from another object it puts a “demon” on the blackboard that will notify it whenever that type of hypothesis is posted. In the case of road shoulder, the external events that it seeks are adjacent road hypotheses with confidence levels of ‘strong-belief,’ ‘belief,’ or ‘partial-support.’

Next we consider the basis of the mapping from endorsements to global confidence values for the shoulder schema. For these experiments, the shoulder schema was designed with the key endorsement of near-road and secondary endorsements of color and texture matches. There are three sets of endorsements that map an internal hypothesis to a global hypothesis with a confidence level of 'belief':

{near-partial-support-road, correct-shoulder-color, correct-shoulder-texture}
{near-believed-road, correct-shoulder-color, correct-shoulder-texture} and
{near-believed-road, neutral-shoulder-color, correct-shoulder-texture}.

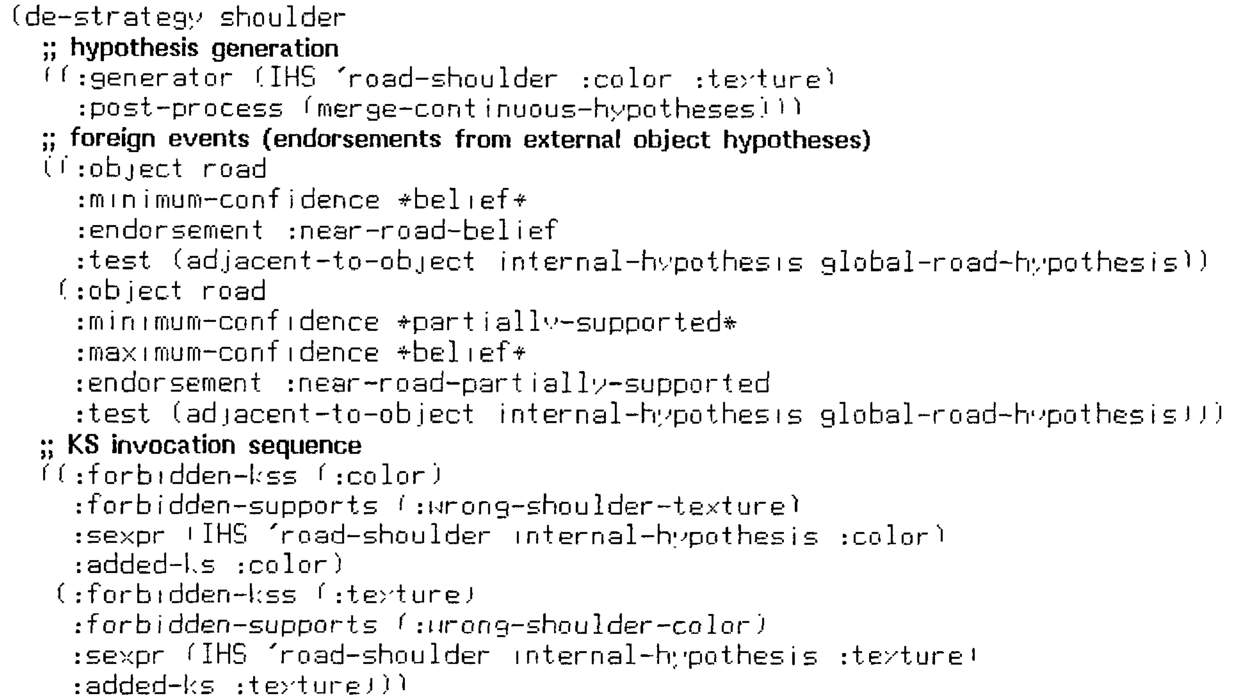
The endorsements for 'strong-belief' are

{near-believed-road, correct-shoulder-color, correct-shoulder-texture}.

No shoulder hypothesis with a negative endorsement for color or texture can score higher than 'slim-evidence,' while most other (nonempty) combinations map to 'partially-supported.'

In addition to the endorsement space and the confidence-mapping function, the schema must specify the control strategy for invoking knowledge sources, and the global hypotheses of other objects that should be read to supply local endorsements. A strategy is specified as a sequence of KS invocations, each of which has a list of required or forbidden endorsements. The road shoulder schema has only one strategy, shown in figure 14. This strategy will not test a hypothesis for a texture match if it already has the negative endorsement *wrong-shoulder-color*, and vice versa.

Let us consider a more complicated schema. The roadline schema is designed to recognize roadlines from the driver's perspective of looking down the road. Roadlines appear in images as yellow or white ribbons (i.e., narrow elongated regions bounded by parallel lines) with little internal texture. They can be extracted (using the low-level knowledge sources listed in section 4) either as regions or as straight lines. Unfortunately, the available region-segmentation algorithms usually perform poorly on narrow objects, and the straight-line extraction algorithms produce



```
(de-strategy shoulder
  ;; hypothesis generation
  ((:generator (IHS 'road-shoulder :color :texture)
    :post-process (merge-continuous-hypotheses)))
  ;; foreign events (endorsements from external object hypotheses)
  ((:object road
    :minimum-confidence *belief*
    :endorsement :near-road-belief
    :test (adjacent-to-object internal-hypothesis global-road-hypothesis))
   (:object road
    :minimum-confidence *partially-supported*
    :maximum-confidence *belief*
    :endorsement :near-road-partially-supported
    :test (adjacent-to-object internal-hypothesis global-road-hypothesis)))
  ;; KS invocation sequence
  ((:forbidden-kss (:color)
    :forbidden-supports (:wrong-shoulder-texture)
    :sexpr (IHS 'road-shoulder internal-hypothesis :color)
    :added-ks :color)
   (:forbidden-kss (:texture)
    :forbidden-supports (:wrong-shoulder-color)
    :sexpr (IHS 'road-shoulder internal-hypothesis :texture)
    :added-ks :texture)))
```

Fig. 14. Road shoulder strategy. "Foreign Events" derive endorsements from global hypotheses. Minimum and maximum confidence fields are used to select hypotheses on the blackboard; the ":test" predicate determines if the hypotheses meets more specific requirements. Forbidden-supports abort a KS invocation sequence; forbidden-kss cause a step to be skipped.

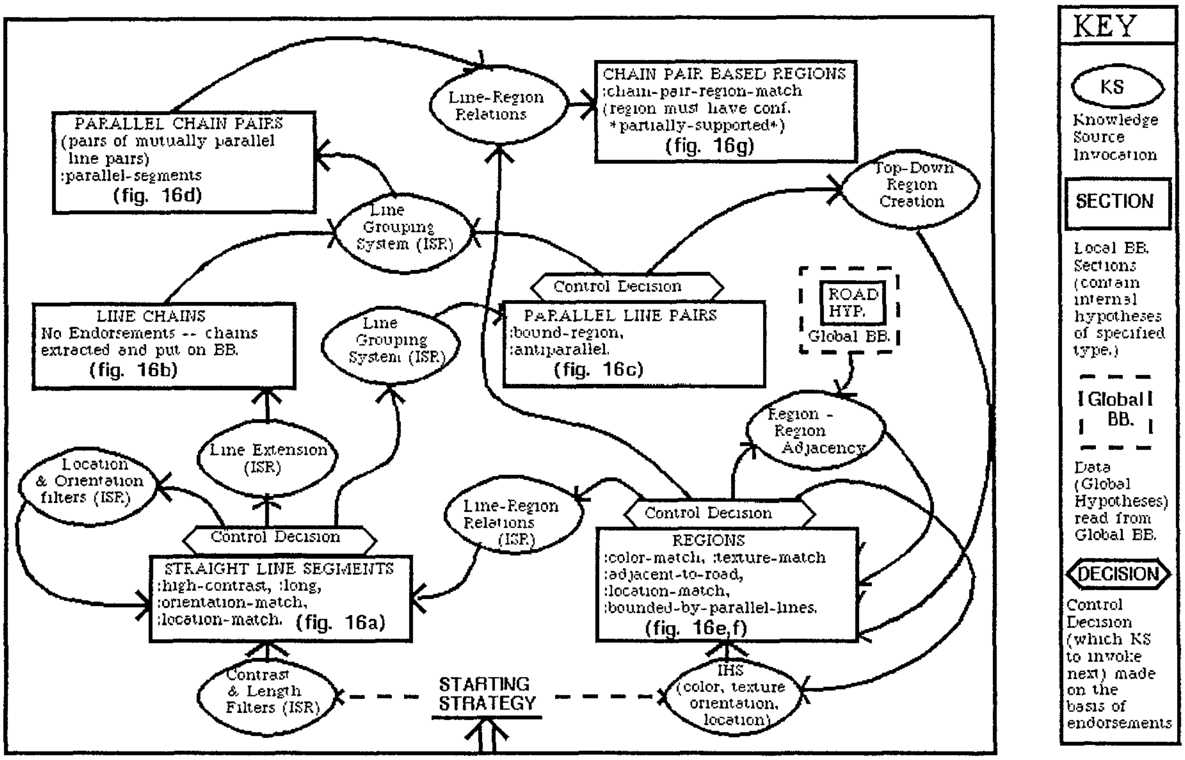
fragments when the roadline bends. Therefore, lines and regions, as well as parallel line pairs, straight-line chains, and pairs of parallel-line chains are used to recognize roadlines. The roadline schema's local blackboard is divided into six sections, one for each type of token (line, region, line pair, line chain, line-chain pair, and final hypothesis). Each type of token, in turn, has a set of possible endorsements that are meaningful for it.

Figure 15 is a graphic representation of the roadline schema. Each level of the local blackboard holds internal hypotheses with a specific type of token (e.g., region, line) and is represented as a rectangle. The type of token and the endorsements defined for it are written inside the rectangle. The knowledge sources, on the other hand, are shown as ellipses; KSs with two incoming arcs compute relations between two internal hypotheses. The control boxes on top of each blackboard section are to remind the reader that the control decision of which KS to execute next on a given internal hypothesis on the local BB is based on the current set of endorsements.

In general, there are two ways in which evidence can be accumulated to support a roadline hypothesis:

1. line data can be grouped into elongated linear structures, and then the color and texture attributes within the lines checked;
2. regions with proper color and texture are used to focus the grouping of straight lines.

In the first case, the system begins by considering all the long, high-contrast lines in the image.These are then tested for the proper orientation (a wide range of orientation's are acceptable, but in a view down the road, the roadline should not be horizontal) and location (the upper third of the image is assumed to be above the horizon and vetoed). Those lines that aquire orientation and location endorsements are put on the local blackboard's straight-line section as internal


PARALLEL CHAIN PAIRS
(pairs of mutually parallel line pairs)
:parallel-segments
(fig. 16d)
Line-Region Relations
CHAIN PAIR BASED REGIONS
:chain-pair-region-match
(region must have conf.
partially-supported)
(fig. 16g)
Line Grouping System (ISR)
Top-Down Region Creation
Control Decision
LINE CHAINS
No Endorsements -- chains extracted and put on BB.
(fig. 16b)
Line Grouping System (ISR)
PARALLEL LINE PAIRS
:bound-region,
:antiparallel.
(fig. 16c)
ROAD HYP.
Global BB.
Region - Region Adjacency
Location & Orientation filters (ISR)
Line Extension (ISR)
Line-Region Relations (ISR)
Control Decision
Control Decision
STRAIGHT LINE SEGMENTS
:high-contrast, :long,
:orientation-match,
:location-match. (fig. 16a)
REGIONS
:color-match, :texture-match
:adjacent-to-road,
:location-match,
:bounded-by-parallel-lines.
(fig. 16e,f)
Contrast & Length Filters (ISR)
STARTING STRATEGY
IHS (color, texture orientation, location)
KEY
KS
Knowledge Source Invocation
SECTION
Local BB. Sections (contain internal hypotheses of specified type.)
Global BB.
Data (Global Hypotheses) read from Global BB.
DECISION
Control Decision (which KS to invoke next) made on the basis of endorsements

Graphical representation of internal hypotheses and data in the roadline schema. Note: (ISR) implies that the knowledge source accessed the ISR database in addition to the local blackboard for data. Colons are used to indicate endorsement names.

Fig. 15. Roadline strategy.

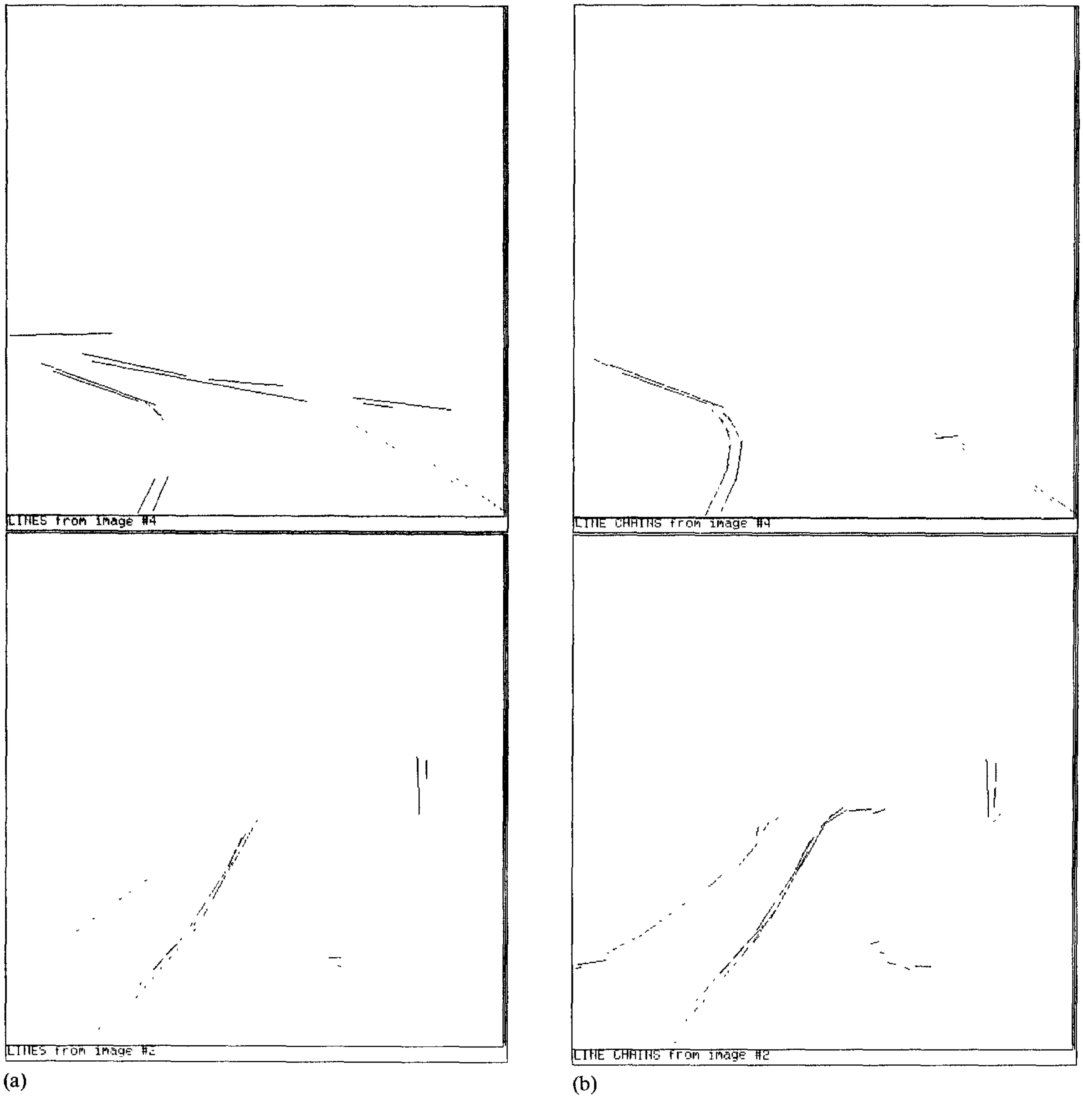
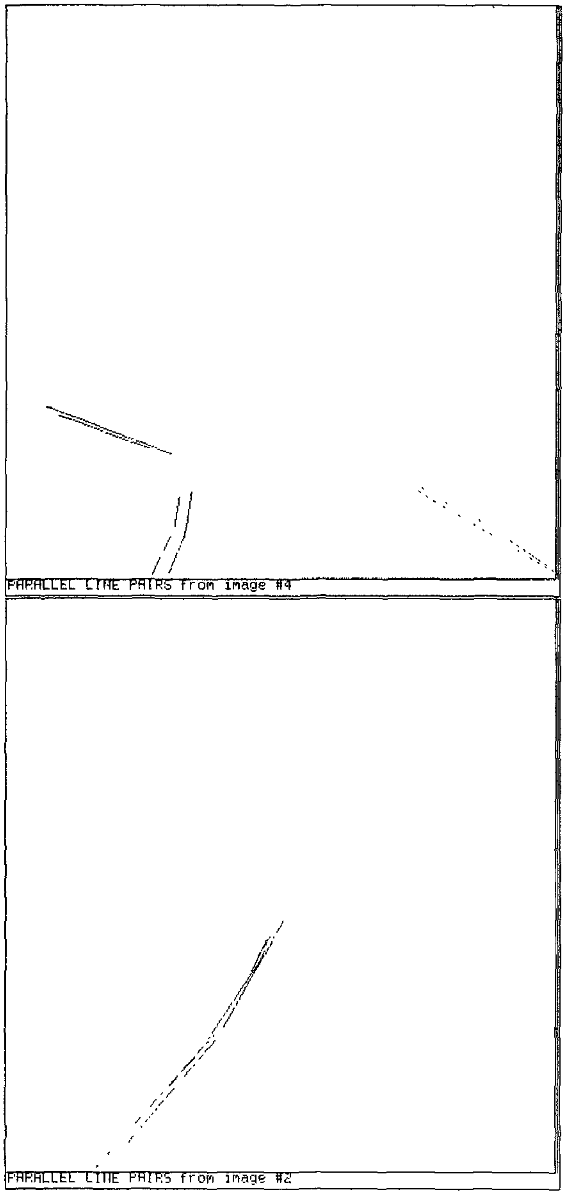
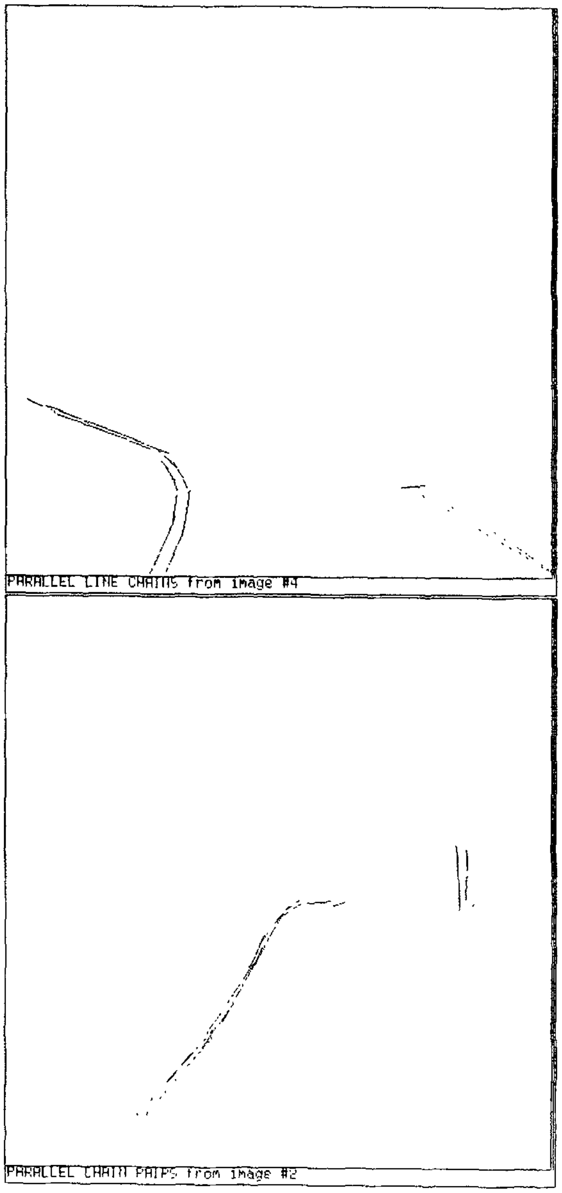


Fig. 16. Data extracted during roadline strategy (refer to blackboard sections of figure 15).

hypotheses (see figure 16a). The line-extension KS extends them into line chains as far as possible, using other (smaller and lower-contrast) lines from the ISR data base (figure 16b). These line chains can also be thought of as piecewise curves. Whenever two line chains have mutually parallel components (see figure 16c), a line-chain pair is made (figure 16d). The chains in a parallel-chain pair are now good candidates to be the edges of a roadline. For each set of parallel lines contained in a chain pair, the ISR is searched to see if these two lines bound a region. If not, a region is made corresponding to the interstitial pixels (figure 16e). The regions are then tested to see if one can be found that matches the expected color (yellow or white) and texture (Low) of roadlines. If so, this


PARALLEL LINE PAIRS from image #4
PARALLEL LINE PAIRS from image #2

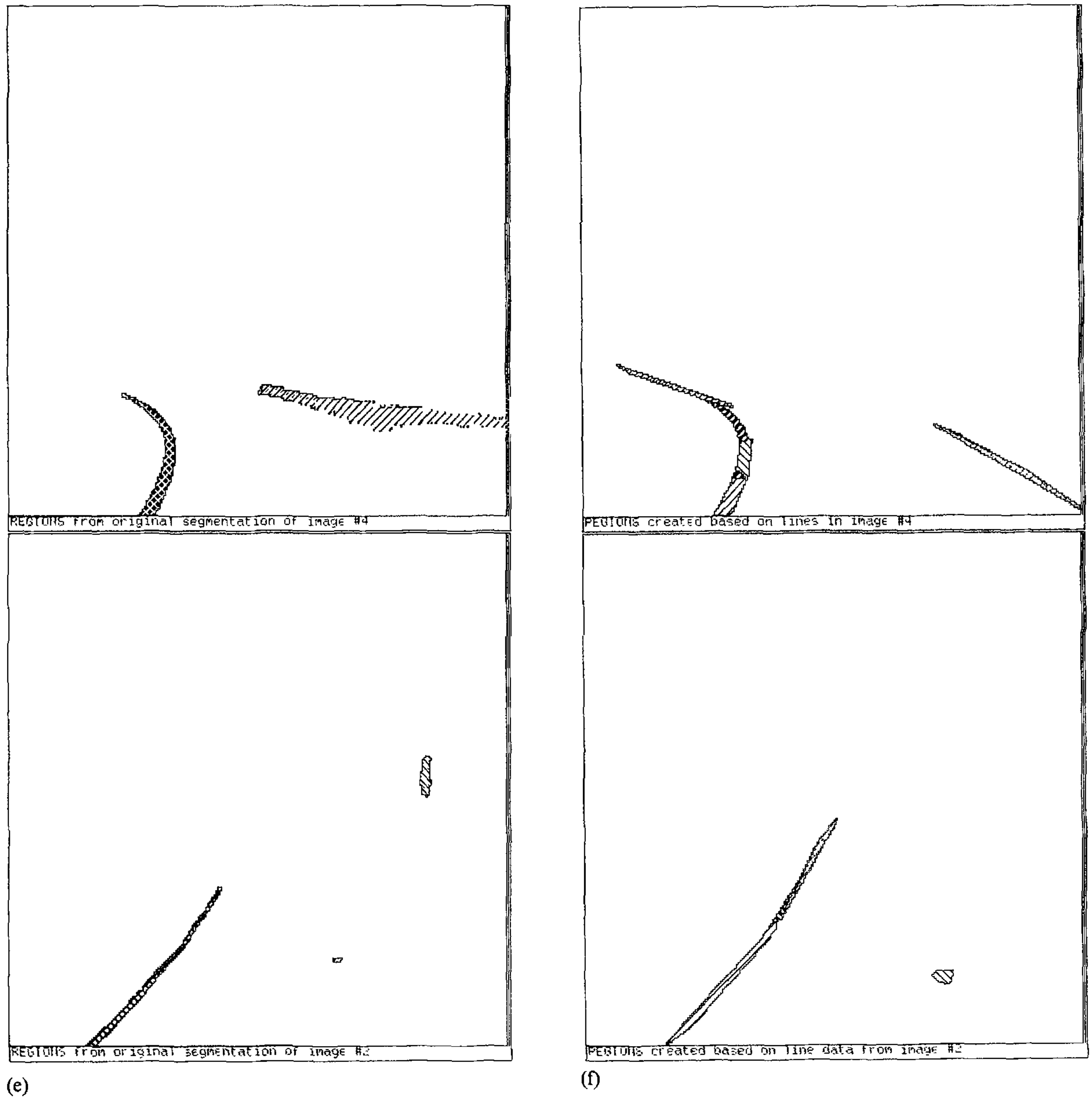
(c)


PARALLEL LINE CHAINS from image #4
PARALLEL CHAIN PAIRS from image #2

(d)

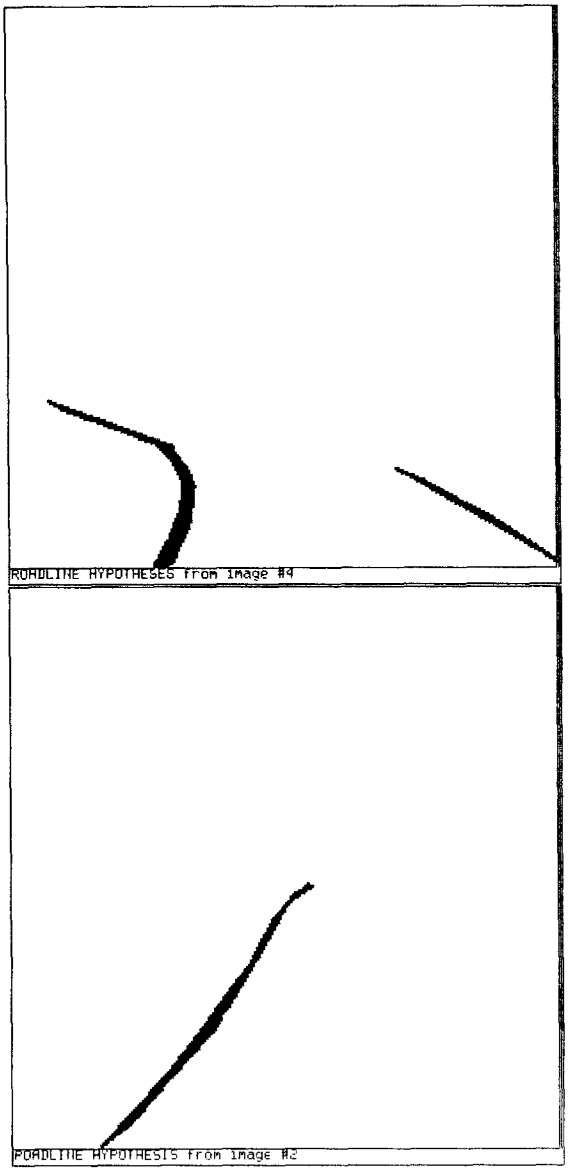
match is used as verification that the parallel-chain pair bounds a roadline, and a new region is made corresponding to the area bounded by the two line chains for as long as they stay parallel (figure 16g). This region serves as the roadline hypothesis.

The method above is very effective when the roadline is straight enough to produce long, high-contrast lines. Indeed, as a result of the line extension KS, it is often able to follow the roadline as it recedes into the background. Unfortunately, highly curved roadlines give it a problem, and our curved-line algorithm was not yet available at the time of these experiments. Thus the second strategy begins by searching for a vertically oriented region that matches the expected color and


REGIONS from original segmentation of image #4
REGIONS from original segmentation of image #2
(e)
REGIONS created based on lines in image #4
REGIONS created based on line data from image #2
(f)

texture of roadline (figure 16f). When such a region is found, any lines that bound it (even very short ones) are candidates for grouping into piecewise curves using the line-extension KS. Once the curves have been found, recognition proceeds as above, except that the lines are already supported by a matching region, and so none has to be created. This method allows the roadline schema to key off the region segmentation as well as straight lines.

Table 3 defines the confidence function for mapping the endorsement of internal hypotheses onto global confidence values. In the roadline schema there are three key endorsements (divided into the two subsets labeled A & B) that must be present to achieve one of the two highest global confidence levels: location-match, orientation-match and chain-pair-region-match (i.e., match of color and texture of a region inside parallel-chain pairs). The remaining two subsets


ROADLINE HYPOTHESES from image #4
ROADLINE HYPOTHESIS from image #2

(g)

of endorsements are used for adjusting the lower confidence levels.

6 Experimental Results

This section demonstrates the Schema System's current capabilities on four road scenes from the Amherst, Mass., area. The results are depicted in figures 18 through 21, and are discussed briefly below. Interpretations of three house scenes generated using an earlier knowledge base are also presented. Interpretations are presented as images in which believed hypotheses (i.e., those with a confidence level of 'belief' or 'strong-belief') are labeled according to object. The key is shown in figure 17. This format supports a qualitative analysis of the interpretation as a whole, but omits many interesting details. A complete reporting of an interpretation would include hypotheses with lower confidence levels, the final state of each schema's internal blackboard, and system performance data such as number of global messages written and read.

The experiments were run with the schema system and its knowledge base on a TI Explorer II Lisp Machine. Low-level procedures written in C were executed on a DEC MicroVax II. For data storage and retrieval, compatible versions of the ISR database were run on both the Explorer and the MicroVax, with the two machines communicating via a Chaosnet link. Interpretations took on the order of 1-2 hours, not including the initial segmentation and line-extraction procedures. It should be noted that no effort to optimize the current system has been made, and that we expect to reduce the interpretation time considerably in the future.

Most instances of objects known to the system have been identified. Large portions of figure 20 are uninterpreted, but this corresponds mostly to barn, an object that was not in the knowledge base. On the one hand, uninterpreted areas demonstrate the obvious disadvantage of knowledge based vision - the system can only recognize objects that are in its knowledge base., On the other hand, we feel that it is better to recognize that a portion of the image is unidentifiable than to apply an incorrect label.

The most serious omission is the left-hand roadline in figure 20. Although the roadline schema is able to create the correct hypothesis (by using HALLUC to create the region from the line data), it is unable to garner enough support for it. This could be corrected with a stronger geometric model of road, which demands that if a right-hand roadline is present, then a left-hand one must be present also. The model used in this experiment, however, implied that all roadlines were optional.

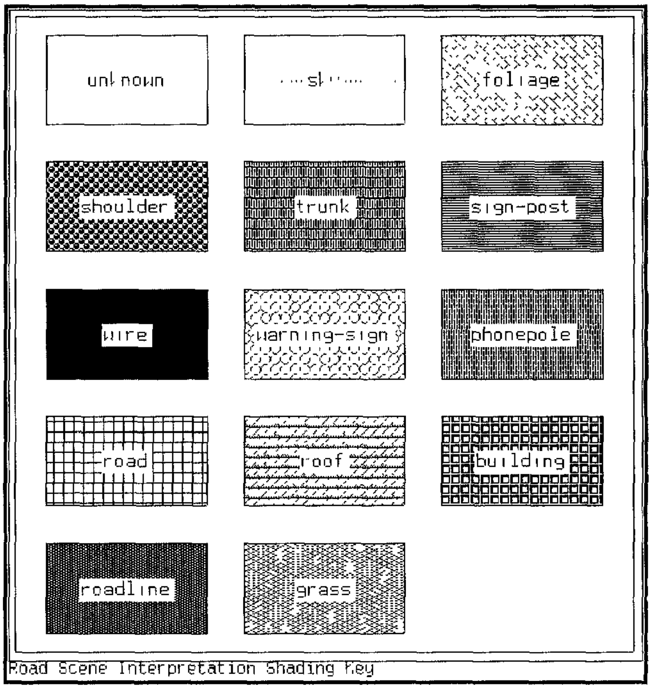


Fig. 17. Pattern key for road scene interpretations.

By looking at intermediate stages of processing, we can see how knowledge was used in interpreting these images. A few hypotheses, such as the road hypotheses in figures 20 and 21 were suggested in their final form by the IHS knowledge source, based on color and texture. These hypotheses were later verified by the presence of parts (roadline and/or shoulder). Most hypotheses, however, go through more transformations. Figure 22 shows the initial feature-based road hypothesis for figure 19 in black. The grey portion shows the area that was added to the road hypothesis in response to the extended roadline hypotheses. This process of using feature classification techniques to focus the attention of a knowledge-directed system onto a limited area, which then extends and refines the hypothesis, is a common technique that is effective across many objects in our experiments.

Figure 23 shows an alternate, "filled in" version of the interpretation in figure 19. This figure shows the strongest hypothesis for each part of the image, even if that hypothesis had a confidence level of less than 'belief.' Hypotheses that do not reach the confidence level 'belief' are interesting for analyzing the effectiveness of the knowledge base. Correct hypotheses with weaker confidence indicate that the schema did not have enough knowledge to verify the hypothesis. In contrast, the absence of a hypothesis indicates that the system never generated the correct internal hypothesis. Some hypotheses of low confidence that are *in*correct are inevitable; an excess of such hypotheses, however, indicates that the system is searching too large a space, and thus more control knowledge is needed.

Figure 20 also shows why telephone wire hypotheses are difficult to generate and need to be spatially constrained. In the road scene domain, wires usually appear as a pair of parallel lines, less than a pixel apart. From many common angles (e.g., looking down the road) they curve, causing straight-line extraction algorithms to produce a series of piecewise linear fragments, rather than a single pair of lines. In addition, thin regions such as these will be composed primarily of mixed pixels, and generally the intensity difference with surrounding areas is very small (at

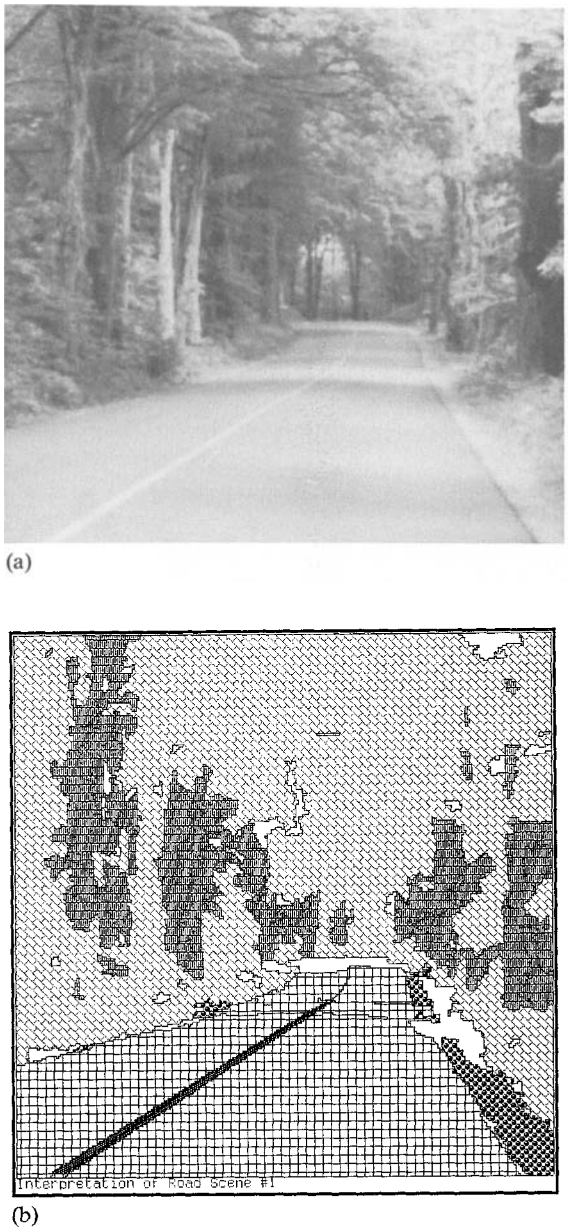


Fig. 18. Road scene #1. (a) Original color image. (b) Final interpretation.

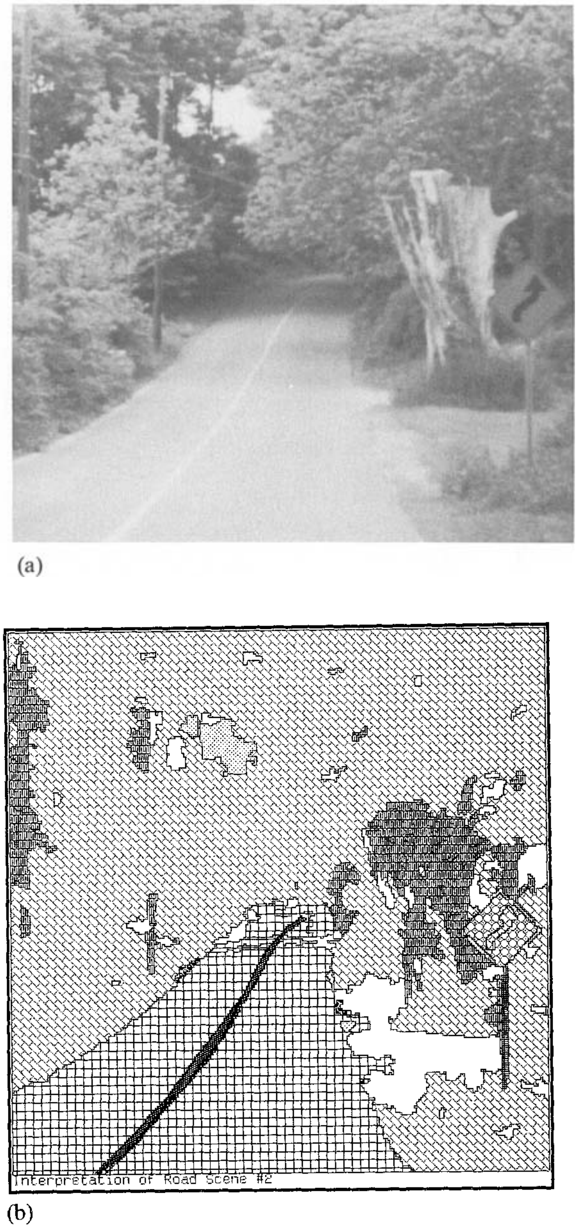


Fig. 19. Road scene #2. (a) Original color image. (b) Final interpretation.

times, less than 4 or 5 intensity levels out of 255). If the entire image were searched for parallel-pairs with slightly darker interstitial pixels, the number of wire hypotheses would be enormous. For the telephone wire schema to be effective, therefore, it is crucial that spatial constraints be provided by the sky and telephone pole schemas.

6.1 House Scene Results

The results in this section were generated using an older house scene knowledge base, the part-of hierarchy of which was shown in figure 12. We present them here to demonstrate that the Schema System has been applied in two different

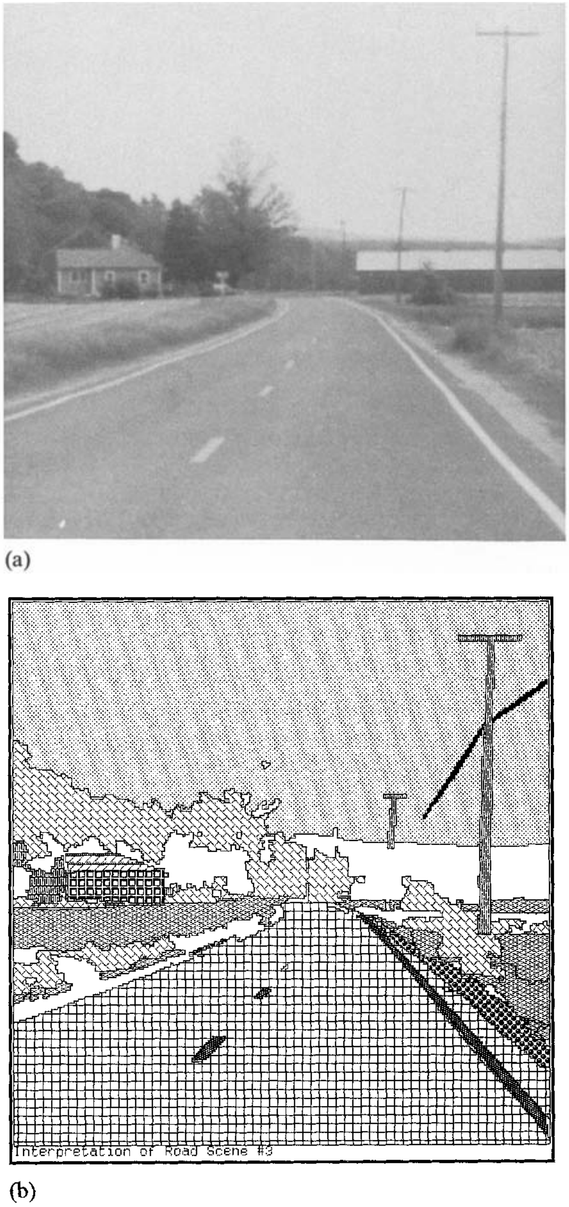


Fig. 20. Road scene #3. (a) Original color image. (b) Final interpretation.

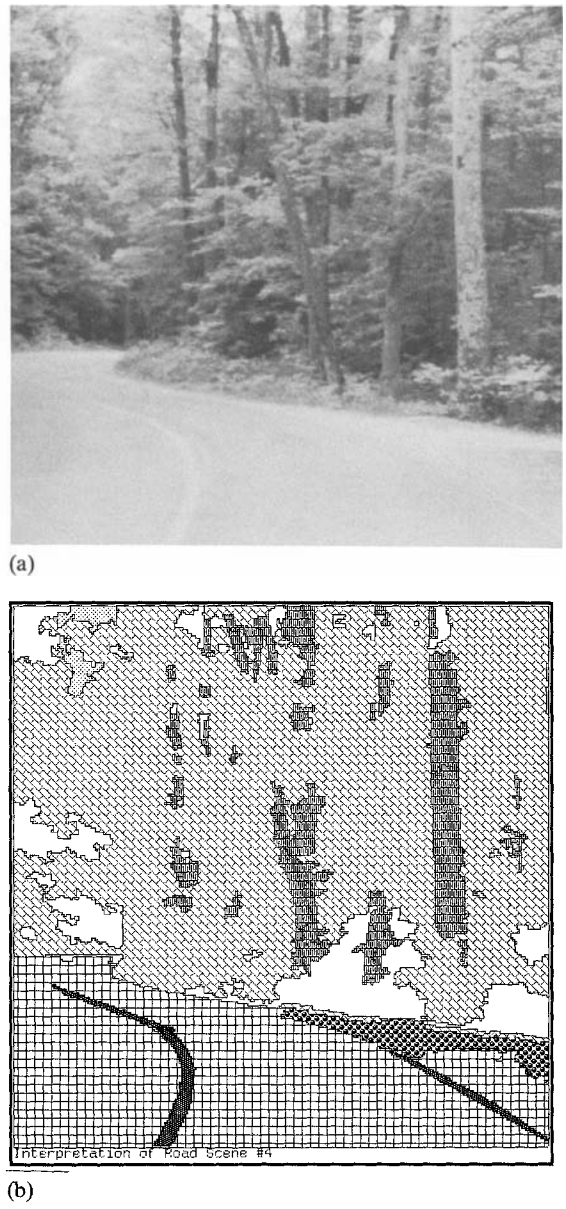


Fig. 21. Road scene #4. (a) Original color image. (b) Final interpretation.

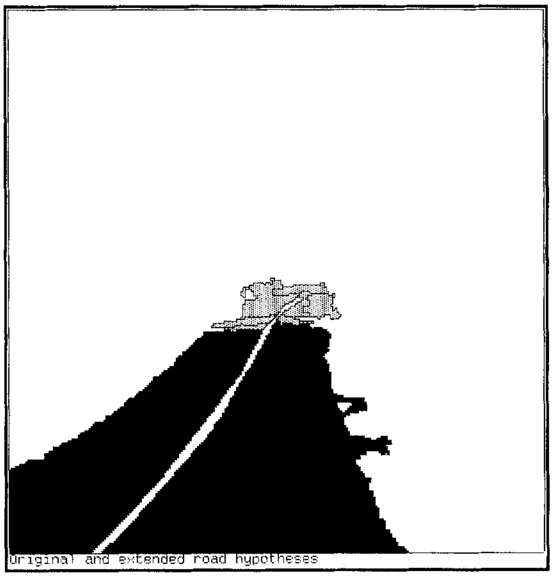


Fig. 22. Original road hypothesis (black) with roadline-based extensions (grey).

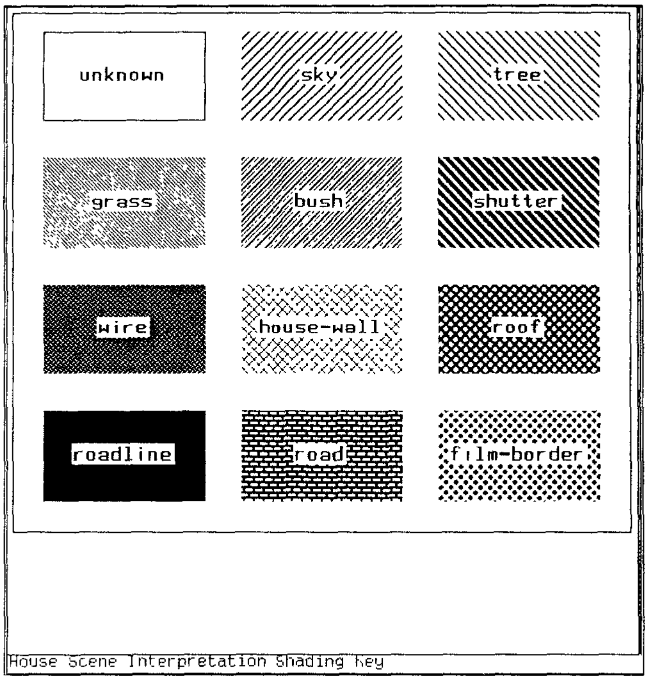


Fig. 24. Pattern key for house scene interpretations.

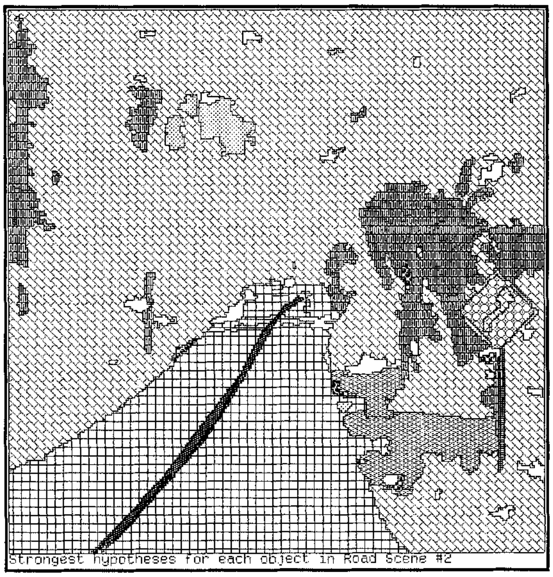


Fig. 23. "Weak" interpretation of road scene #2. Best hypothesis for each region.

domains. A key for the house scene interpretations that appear in figures 25 through 27 is shown in figure 24. It should be noted that the house scene results are two years old at the time of writing, and that significant progress has taken place in the Schema System since then. By way of contrast, the road scene knowledge base was more quickly assembled than the house scene data base, and is less brittle. Moreover, there are some qualitative differences. The knowledge sources available at the time that the house scene results were obtained were primarily region based. As a result, the system had difficulty recovering from incorrect window and shutter segmentations (as in figure 27). In addition, the top-down creation of tokens was difficult in the software environment utilized at that time. As a result token creation was only performed for roofs and telephone wires. Today, the resulting interpretations could be improved using hallucinated tokens. At some point in the future, the house scene knowledge base will be updated and combined with the road scene schemas.

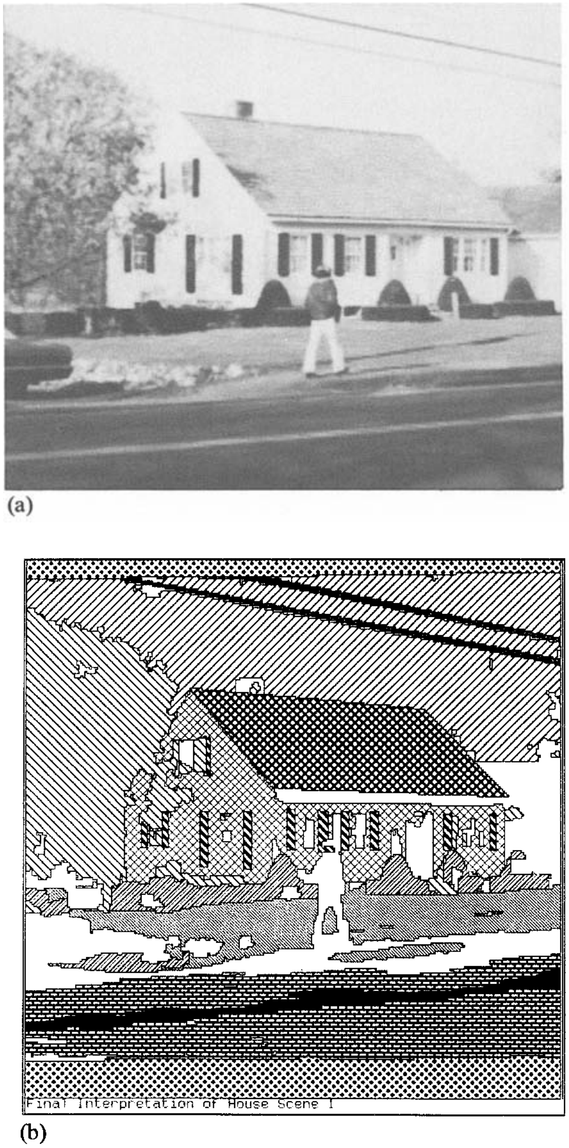


Fig. 25. House scene #1. (a) Original color image. (b) Final interpretation.

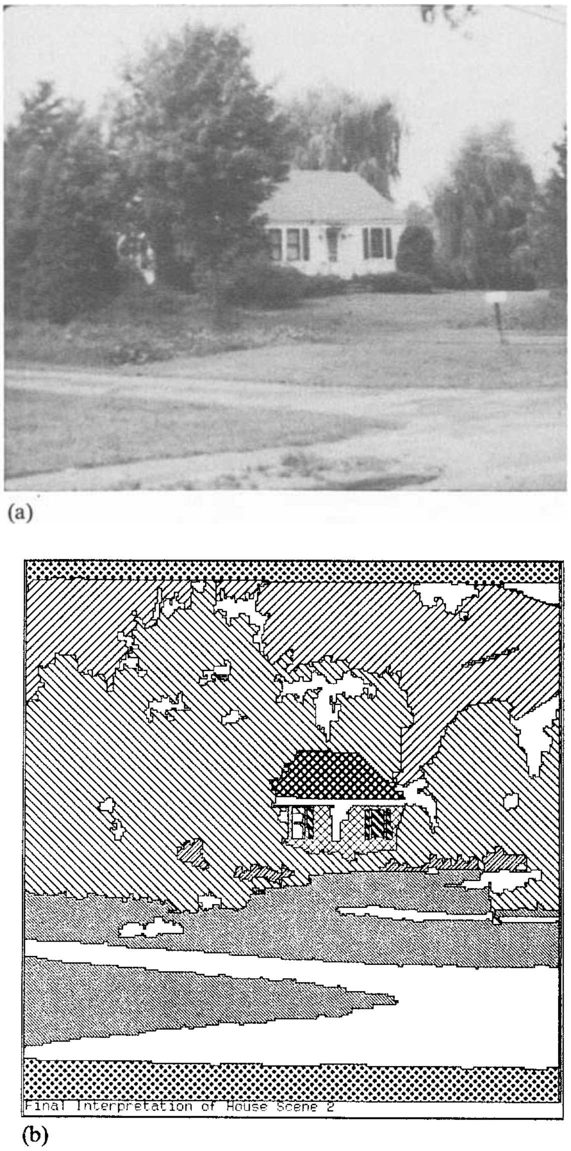


Fig. 26. House scene #2. (a) Original color image. (b) Final interpretation.

7 Conclusion and Future Work

7.1 Summary

The UMass Schema System is an exercise in knowledge-directed vision, emphasizing the control aspect of image interpretation. The goal is to provide the breadth of a general-purpose vision system while retaining the efficiency of a special-purpose one. The approach is to build generic mechanisms that avoid incurring the cost of complete generality when object-based expectations can be used to constrain visual processing. To this end, the system has been designed to provide a flexible environment for encoding both object and control knowledge. Assembled within an object-specific schema is knowledge about which knowledge sources and representations are appropriate, when they should be used, and how the

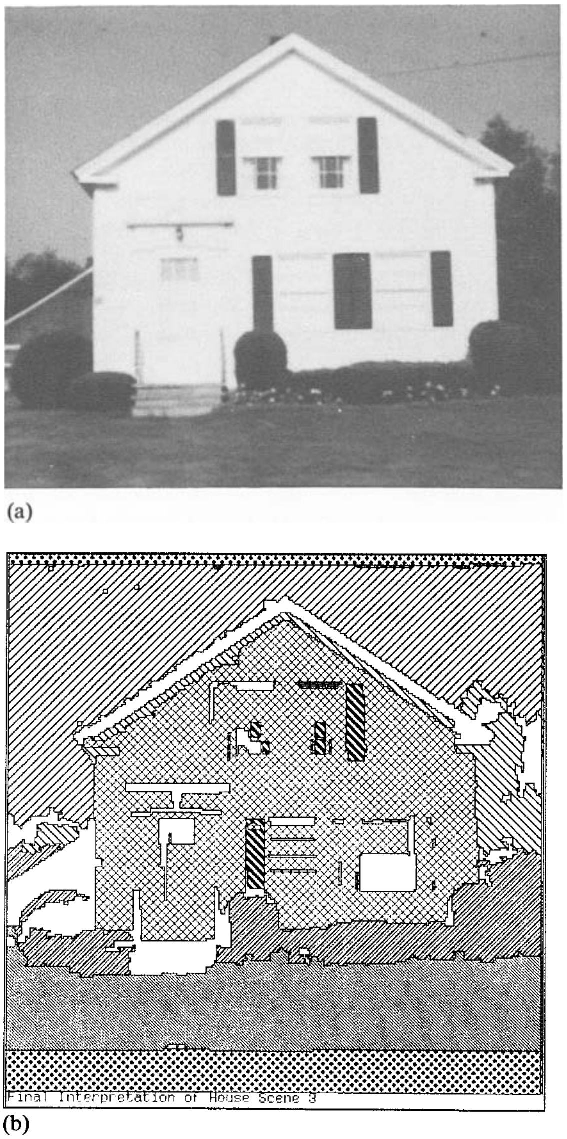


Fig. 27. House scene #3. (a) Original color image. (b) Final interpretation.

evidence provided by each should be combined.

Each schema is an expert at recognizing one type of object. In an attempt to exploit course-grained parallelism, each instantiated schema runs in parallel with other schemas, and communicates with them asynchronously through a global blackboard. Together, the set of running object schemas cooperate to interpret the scene. Further parallelism is provided within each schema by the use of multiple *strategies*. Strategies are simple control programs which encode knowledge about what knowledge should be used and how to evaluate partial hypotheses. The Schema System architecture is an extension of the blackboard paradigm into a distributed environment that eschews centralized control mechanisms in favor of intelligent and autonomous schemas.

In the research reported here, we have chosen to embed spatial constraints into prototypical views of the class of scenes with which we are experimenting. However, during this development we did not adhere to a rigid methodology, and therefore some schemas used strong constraints that allowed fairly specific relational information to other object parts in the schema; in other cases there were weaker assumptions that allowed the schema to be applied to a wider range of viewpoints at the expense of not being able to apply spatial reasoning, or to do so in a less constrained manner.

7.2 Future Research

Probably the most important lesson for us was one that was anticipated. A partially declarative approach to schema construction allowed us to rapidly develop a complex system in a flexible manner while minimizing the construction of general machinery. Based on our experience, it is clear that the process of constructing object and scene schemas, i.e., the knowledge-engineering process, will greatly benefit from a more declarative representation. This includes the specification of confidence-mapping functions and constraints on the acceptable values of token attributes and relations. We believe that this will allow a far more rapid cycle of knowledge representation and experimentation. Note, however, that in contrast to other methodologies, such as production rule systems, we are making the specification of control a key part of the knowledge base. In particular, we are not using a general monolithic inference and control engine for applying the knowledge, but rather providing the means for a user to define customized control behaviors declaratively.

From its inception, the Schema System has been designed to run on a parallel processor. Care has been taken to distribute the vision task while avoiding communication bottlenecks. However, until the Schema System is running efficiently on a parallel processor we cannot claim to have demonstrated its effectiveness in this environment. Questions will remain, such as whether the cost of process creation will overwhelm the benefits of course-grained parallelism, and can the allocation of a fixed number of processors be handled effectively without introducing a centralized mechanism.

While it is imperative that the Schema System be exercised in a truly parallel environment, our target machine, the IUA [59], will only exist in a scaled-down version in the near future. Therefore, we plan to port the system to a Sequent Balance 21000™ multiprocessor. Although this machine is not adequate for real-time vision, it should allow us to test our schema control framework in a parallel, shared-memory environment.

Another major question for all knowledge-based vision systems is how well can the knowledge base expand to accommodate new objects and domains. The Schema System must become sufficiently good at object indexing that only a few, relevant schemas, associated with appropriately constrained viewpoints, are ever invoked on any given image. The prediction hierarchy of Burns may help here. In addition, the effects of knowledge base size on communication patterns, control, etc., must be determined. Unfortunately, our current knowledge bases of 15 objects each are inadequate for such an investigation. Serious research into the construction and maintenance of large knowledge bases can probably begin when the system is able to recognize 50 to 100 objects.

With regard to task domains, we plan to integrate our house and road scene knowledge bases in the near future. Since these domains have many objects in common, the resulting data base should contain around 20 objects, and enable the system to recognize objects from a wider variety of images. Work on aerial images is also being considered.

References

1. M.A. Arbib, "Segmentation, schemas, and cooperative computation," in *Studies in Mathematical Biology, pt.1*, S. Levin (ed.). MAA Studies in Mathematics, vol. 15, pp. 118-155, 1978.
2. M.A. Arbib and A.R. Hanson, eds., *Vision, Brain, and Cooperative Computation*, MIT Press: Cambridge, MA, 1987.
3. D.H. Ballard, "Model-directed detection of ribs in chest radiographs," *Proc. 4th IJCPR*, Kyoto, Japan, 1978.
4. R. Belknap, A. Hanson, and E. Riseman, "The information fusion problem and rule-based hypotheses applied to complex aggregations of image events," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, June 22-26, 1986.
5. J.R. Beveridge, J. Griffith, R.R. Kohler, A.R. Hanson, and E.M. Riseman, "Segmenting images using localized histograms and region merging," *Intern. Computer Vision*, this issue. Also COINS Tech. Rept. 87-88, Univ. of Massachusetts at Amherst, October 1987.
6. T. Binford, "Survey of model-based image analysis systems," *Intern. J. Robotics Res.* 1: 18-64, 1982.
7. M. Boldt and R. Weiss, "Token-Based Extraction of Straight Lines," COINS Tech. Rept. 87-104, Univ. of Massachusetts at Amherst, October 1987.
8. R. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artifical Intelligence* 17: 285-348, 1981.
9. J.B. Burns, A.R. Hanson, and E.M. Riseman, "Extracting straight lines," *IEEE Trans. PAMI-8(4)*, pp. 425-455, July 1986.
10. J.B. Burns and L.J. Kitchen, "Recognition in 2D images of 3D objects from large model bases using prediction hierarchies," *Proc. 10th Intern. Joint Conf. Artif. Intell.*, Milan, pp. 763-766, August 1987.
11. J.B. Burns, *Recognition in 2D Images of 3D Objects from Large Model Bases Using Prediction Hierarchies*. Forthcoming doctoral thesis, Univ. of Massachusetts at Amherst.
12. J.H. Burrill, *Low Level Vision System*, COINS Tech. Rept. 87-14, Univ. of Massachusetts at Amherst, January 1987.
13. D. Chapman, "Planning for conjunctive goals." *Artificial Intelligence*, 29 333-377, July 1987.
14. P. Cheeseman, "An inquiry into computer understanding," *Computational Intelligence* 4(1): 58-66, February 1988.
15. P.R. Cohen. "Numeric and Symbolic Reasoning About Uncertainty in Expert Systems," COINS Tech. Rept. 85-25, Univ. of Massachusetts at Amherst, 1985.
16. D. Corkill, K. Gallagher, and K. Murray. "GBB: A generic blackboard development system" *Proc. of AAAI-86*, Philadelphia, PA, vol. 2, pp. 1008-1014, 1986.
17. J. Dolan, G. Reynolds, and L. Kitchen, "Piecewise circular description of image curves using constancy of grey-level curvature," COINS Tech. Rept. 86-33, Univ. of Massachusetts, July 1986.

18. J. Doyle, "A truth maintenance system," *Artificial Intelligence* 12: 231-272, 1979.
19. B.A. Draper, R.T. Collins, J. Brolio, J. Griffith, A.R. Hanson, and E.M. Riseman, "Tools and experiments in the knowledge-directed interpretation of road scenes," *Proc. DARPA Image Understanding Workshop*, Los Angeles, pp. 178-193, February 1987.
20. L.D. Erman, F. Hayes-Roth, V.R. Lesser, and D.R. Reddy, "The Hearsay-II Speech-Understanding System: Integrating knowledge to resolve uncertainty," *Computing Surveys* 12: 213-253, June 1980.
21. E.A. Feigenbaum, A. Barr, and P. Cohen, (eds.), *The Handbook of Artificial Intelligence*. William Kaufmann: Los Altos, CA, 1981.
22. J. D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*. Addison-Wesley: Reading, MA, 1982.
23. Z. Gigus and J. Malik, "Computing the aspect graph for line drawings of polyhedral objects," Report UCB-CSD 88-402, Univ. of California, Berkeley, February 1988.
24. J. Glicksman, "A cooperative scheme for image understanding using multiple sources of information," Tech. Rept. TN82-13 (Ph.D. thesis), Dept. of Computer Science, Univ. of British Columbia, November 1982.
25. A.R. Hanson and E.M. Riseman, "VISIONS: A computer system for interpreting scenes." In *Computer Vision Systems*, Hanson and Riseman (eds.), Academic Press: New York, 1978.
26. A.R. Hanson and E.M. Riseman, "The VISIONS image understanding system—1986," COINS Tech. Rept. 86-62, Univ. of Massachusetts at Amherst, December 1986 and In *Advances in Computer Vision* C. Brown (ed.), Erlbaum: Hillsdale, NJ, 1987.
27. A.R. Hanson and E.M. Riseman, "From image measurements to object hypotheses," COINS Tech. Rept. 87-129, Univ. of Massachusetts at Amherst, December 1987.
28. B. Hayes-Roth, "A blackboard architecture for control," *Artificial Intelligence* 26(3): 250-321, July 1985.
29. S.V. Hwang, "Evidence accumulation for spatial reasoning in aerial image understanding." Ph.D. thesis, Dept. of Computer Science, Univ. of Maryland, 1984.
30. K. Ikeuchi, "Precompiling a geometrical model into an interpretation tree for object recognition in bin-picking tasks," *Proc. DARPA Image Understanding Workshop*, Los Angeles, pp. 321-339, February 1987.
31. P.M. Johnson, D.D. Corkhill, and K.Q. Gallagher, "Integrating BB1-style control into the generic blackboard system," Presented at the AAAI Workshop on Blackboard Systems, Seattle, Washington, July 1987.
32. R.M. Knutson. *Flattened Fauna: A Field Guide to Common Animals of Roads, Streets, and Highways*, Ten Speed Press: Berkeley, CA, 1987.
33. J.J. Koenderink and A.J. van Doorn, "The singularities of the visual mapping," *Biological Cybernetics* 24(1): 51-59, 1976.
34. C.A. Kohl, "GOLDIE: A goal-directed intermediate-level executive for image interpretation," COINS Tech. Rept. 88-22, (Ph.D. thesis) Dept. of Computer and Information Science, Univ. of Massachusetts at Amherst, 1988.
35. C.A. Kohl, A. Hanson, and E. Riseman. "A goal-directed intermediate level executive for image interpretation," *Proc. 10th Intern. Joint Conf. Artif. Intell.*, Milan, pp. 811-814, August 1987.
36. R.R. Kohler, "Integrating non-semantic knowledge into image segmentation processes," COINS Tech. Rept. 84-04, (Ph.D. thesis) Dept. of Computer and Information Science, Univ. of Massachusetts at Amherst, 1984.
37. R.E. Korf., "Macro-operators: A weak method of learning," *Artificial Intelligence* 26: 35-77, 1985.
38. N.B. Lehrer, G. Reynolds, and J. Griffith, "Initial hypothesis formation in image understanding using an automatically generated knowledge base," *Proc. DARPA Image Understanding Workshop*, Los Angeles, pp. 521-537, February 1987. Also, COINS Tech. Rept. 87-04, Univ. of Massachusetts at Amherst, January 1987.
39. D.M. McKeown Jr., W.A. Harvey, Jr., and J. McDermott, "Rule-based interpretation of aerial imagery," *IEEE Trans. PAMI-7(5)*: 570-585, 1985.
40. M. Minsky, "A framework for representing knowledge," MIT AL Laboratory Memo 306, 1974. Also in *Psychology of Computer Vision*, P.H. Winston (ed.), McGraw-Hill: New York, 1975.
41. M. Minsky, "The society theory of thinking." In *Artificial Intelligence: An MIT Perspective*, P.H. Winston and R.H. Brown (eds.), vol. 1. pp. 423-450, MIT Press: Cambridge, 1979.
42. M. Nagao and T. Matsuyama, *A Structural Analysis of Complex Aerial Photographs*. Plenum Press: New York, 1980.
43. P.A. Nagin, A.R. Hanson, and E.M. Riseman, "Studies in global and local histogram-guided relaxation algorithms," *IEEE Trans. PAMI-7(5)*: 263-277, 1985.
44. Y. Ohta, "A region-oriented image-analysis system by computer," Ph.D. thesis, Dept. of Information Science, Kyoto University, March, 1980.
45. J. Pearl, "Reverand Bayes on inference engines: A distributed hierarchial approach." *Proc. 2nd Conf. Arti. Intell.* Pittsburgh, PA, pp. 133-136, 1982.
46. H. Plantinga and C. Dyer, "Visibility, Occlusion, and the aspect graph," Report 736, University of Wisconsin-Madison, December 1987.
47. G. Reynolds, J.R. Beveridge, "Searching for geometric structure in images of natural scenes," *Proc. DARPA Image Understanding Workshop*, Los Angeles, pp. 257-271, February 1987. Also COINS Tech. Rept. 87-03, Univ. of Massachusetts at Amherst, July 1986.
48. E.M. Riseman and A.R. Hanson, "A methodology for the development of knowledge-based vision systems," *Proc. IEEE Workshop on Principles of Knowledge-Based Systems*, Denver, December 1984. Also COINS Tech. Rept. 86-27, Univ. of Massachusetts at Amherst, July 1986.
49. E.M. Riseman, A.R. Hanson, and R. Belknap, "The information fusion problem: Forming token aggregations across multiple representations," COINS Tech. Rept. 87-48, Univ. of Massachusetts at Amherst, December 1987.
50. A. Rosenfeld et. al. "DIALOG — 'Expert' vision systems:

Some issues," *Computer Vision, Graphics and Image Processing* 34 99–117, 1986.

51. G. Shafer, *A Mathematical Theory of Evidence* Princeton University Press: Princeton, 1976.
52. S.A. Shafer, A. Stentz, and C.E. Thorpe, "An architecture for sensor fusion in a mobile robot," *Proc. IEEE Inter. Conf. Robotics and Automation*, San Francisco, pp. 2002-2011, 1986.
53. S.C. Shapiro, ed., *The Encyclopedia of Artificial Intelligence.* Wiley: New York. 1987.
54. E. Soloway, J. Bachant, K. Jensen, "Assessing the maintainability of XCON-in-RIME: Coping with the problems of a VERY large rule-base," *Proc. 6th AAAI Nat. Conf. Artif. Intell.*, Seattle, WA, vol. 2, pp. 824–829, 1987.
55. C. Thorpe, S. Shafer, T. Kanade, "Vision and navigation for the Carnegie Mellon Navlab," *Proc. DARPA Image Understanding Workshop*, Los Angeles, pp. 143–152, February 1987.
56. J.K. Tsotsos, "Knowledge organization and its role in representation and interpretation for time-varying data: The ALVEN system," *Computational Intelligence* 1: 16–32, 1985.
57. J.K. Tsotsos, "Image understanding." In *The Encyclopedia of Artificial Intelligence*, S.C. Shapiro, (ed.) New York, 1987.
58. J.K. Tsotos, "A 'Complexity Level' analysis of vision," *Int. J. Computer Vision* 1(4): 303-320.
59. L.W. Tucker, "Computer vision using quadtree refinement," Ph.D. dissertation, Polytechnic Institute of New York, May 1984.
60. C. Weems, S. Levitan, A. Hanson, E. Riseman, D. Shu, G. Nash. "The Image Understanding Architecture," *Int. J. Computer Vision* 2(3), January 1989.
61. R. Weiss and M. Boldt, "Geometric grouping applied to straight lines," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami, June 22–26, 1986.
62. L.P. Wesley, "Evidential-based control in knowledge-based systems," COINS Tech. Rept. 88-24, Ph.D. thesis, Department of Computer and Information Science, Univ. of Massachusetts at Amherst, 1988.
63. T.E. Weymouth, "Using object descriptions in a schema network for machine vision," COINS Tech. Rept. 86-24, (Ph.D. thesis), Department of Computer and Information Science, Univ., of Massachusetts at Amherst, May 1986.
64. P.H. Winston, "Learning structural descriptions from examples," In *Readings in Knowledge Representation*, R.J. Brachman and H.J. Levesque (eds.), Morgan Kaufman Publishers, Inc., 1985. Also in *Psychology of Computer Vision*, P.H. Winston (ed.), McGraw-Hill: New York, 1975.
65. L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems* 1(1), 1978.