Mapsee2's schema-based representations support efficient recognition and search, as well as overcoming some inherent limitations of the well-known network consistency approach to scene analysis.

Representing Knowledge of the Visual World

William Havens and Alan Mackworth, University of British Columbia

The central issue in artificial intelligence—the representation and use of knowledge—unifies areas as diverse as natural-language understanding, speech recognition, story understanding, planning, problem solving, and vision. This article focuses on how computational vision systems represent knowledge of the visual world. It examines current methodology under two criteria: descriptive adequacy, the ability of a representational formalism to capture the essential visual properties of objects and the relationships among objects in the visual world, and procedural adequacy, the capability of the representation to support efficient processes of recognition and search.

A major theme in computational vision has been the distinction between the methodology of image analysis (or early vision) and scene analysis (or high-level vision). Briefly, image analysis can be characterized as the science of extracting from images useful descriptions of lines, regions, edges, and surface characteristics up to the level of Marr's 2½-D sketch.[1] It is generally assumed that image analysis is domain independent and passive, that is, data driven. Scene analysis attempts to recognize visual objects and their configurations. It is viewed as domain dependent and goal driven, motivated by the necessity of identifying particular objects expected to be present in a scene.

Although some may disagree, these distinctions should be seen not as a strict dichotomy but as a spectrum. Early vision exploits constraints that are usually valid in the particular visual world for which it has evolved (or been designed). Although early vision is predominantly data driven, high-level visual processes must be able to establish parameters for and control the attention of lower level processes. As we argue later, efficient scene analysis systems must combine goal-driven and data-driven recognition processes. If that dichotomy is actually a spectrum then establishing the exact boundary is not a research issue.

In this article, we outline current scene analysis methodology (early vision is ably described elsewhere[1,2]) and identify a number of its deficiencies. In response to these problems, some recent systems use schema-based knowledge representations. Examples taken from one called Mapsee2 illustrate our arguments.

Progress in high-level vision

The necessity of adequate representations for visual knowledge has been a constant theme in high-level vision research. The very early work of Roberts[3] established an initial research paradigm that has persisted for 20 years.

Roberts' system consisted of two programs. An image analysis program constructed a line drawing that served as input to his scene analysis program. From a gray-scale image the image analysis line-finder constructed a line drawing using spatial differentiation, clipping, and line-following techniques. The subsequent scene analysis program assumed that the visual world consisted of instances of three simple polyhedral models: a cube, a wedge, and a hexagonal prism. These primitives were allowed to be scaled, translated, and rotated. Composite objects were constructed of instances of the primitives glued together.

The scene analysis program iterated through a cycle of four processes: cue discovery, model invocation, model verification, and model elaboration.[4] A variety of topological image cues used to index into the set of primitive models found candidate matches without exhaustive analysis by synthesis. The model fragment thus invoked was then subjected to metrical tests to judge its fit to the image. If a successful partial fit was obtained,

0018-9162/83/1000-0090$01.00 1983 IEEE

the appearance of the rest of the model was predicted in the image. A good match between the prediction and the image indicated a successful model hypothesis. The predicted appearance of the model was then used to produce a new line drawing of the scene with that portion of the scene deleted from the image. The cycle repeated until the entire image had been accounted for.

Although limited, Roberts' program provided a major impetus to computational vision research,[5] and his *blocks world* approach was the main one for the subsequent decade. The Huffman-Clowes labeling scheme, introduced in the early 1970's, was a crucial breakthrough. Its key ideas are that edge types (convex, concave, and occluding) in the scene domain can be determined from image domain evidence (junction shapes) and that an edge cannot change its type from one end to the other (a scene domain coherence rule). In the cue-model paradigm, a junction shape acts as a cue for a number of corner models in the scene domain. This local ambiguity can be globally reduced by enforcing the edge object coherence rule between adjacent corners. Extending these ideas, Waltz[6] made two contributions. He extended the descriptive adequacy of this scheme by allowing additional edge types such as cracks and shadows. He enhanced the procedural adequacy by introducing a filtering algorithm that removes local inconsistencies before constructing global solutions. He gave some experimental evidence that this could be more efficient than backtracking.

The filtering algorithm has been generalized to a class of formal network consistency algorithms for problems in which a number of variables have to be instantiated in associated domains while satisfying a set of binary constraints.[7] The constraint-based approach to knowledge representation in vision has been applied to other visual domains. Mapsee[8] interprets freehand geographical sketch maps. In this world, image lines or chains can be scene roads, rivers, bridges, mountains, towns, lakeshores, or seashores, while image regions can be land, lake, or ocean. The constraint approach uses these entities as the objects to be instantiated, while the models are derived from scene domain knowledge of how the objects can interact. For example, a T-junction of two image chains could be a road junction or a river junction or a river going under a bridge, etc. The models are thus n-ary constraints on the objects, and the network consistency algorithms are generalized to cope with that extension.

The complexity barrier

The computational paradigm introduced by Roberts and developed by others is now mature. It has resulted in a uniform representational framework for encoding and manipulating knowledge about the visual world. Unfortunately, network consistency has reached its inherent limitations. It does not easily scale upward to more complex domains and exhibits a number of shortcomings:

Limitation 1. The objects defined in the representation correspond only to primitive scene entities. Complex scene interpretations must be expressed solely as atomic labels for these primitive objects. Consequently, abstract high-level scene interpretations are represented only implicitly by projection onto the low-level label sets of the objects and must be reconstructed from the low-level interpretations after the recognition process has terminated. Projecting abstract interpretations onto an object's label set causes set size to grow exponentially with the complexity of the scene domain. This phenomenon was a major obstacle in Waltz's research. We conclude that objects at the lowest level of description in a system are not appropriate hooks for attaching high-level interpretations.

Limitation 2. The models are impoverished. Each model is represented as a relation over the label sets of a small number of neighboring objects in the network and, therefore, can express only local constraints on the scene. No explicit descriptions of the structural relationships appearing in the overall scene are represented. Instead, they are implicit in the relations themselves.

Limitation 3. The extension of the label set for each object has been represented explicitly. Network consistency methods proceed by deleting from the label set of each object any label that does not satisfy every model constraining that object. Any deleted label cannot be part of a global scene interpretation. Label sets are usually represented explicitly as a list of atoms, each naming a particular interpretation. Furthermore, each label must be considered independently, even though many of the labels in a given label set have a partial common interpretation. More efficient, intensional representations for interpretations are needed.

Limitation 4. A compiler must be constructed to compute the label sets for each type of object in the system. This compiler, given a suitable description of the semantics of the scene domain, considers exhaustively all possible scene configurations and represents those configurations in the label sets of the primitive objects.

Limitation 5. Network consistency relies on a single level of cues and models. Cues are image properties computed context-free from the input image. Once discovered, they are used to invoke appropriate models directly. Since each model depicts relationships among objects at a single level of abstraction, its semantics must be tied closely to the invoking image cue. Therefore, models for high-level abstract scene relationships are not possible. Attempts at using low-level image cues to invoke high-level models have been disappointing.[2] What is needed is a hierarchy of cues and models. Low-level, context-free cues should be used to invoke low-level scene models, and high-level, context-sensitive cues, which have been computed as a result of recognition, should be used to invoke high-level models.[9]

Limitation 6. Procedural knowledge is absent. Network consistency employs a uniform constraint propagation control structure to guide the search process. Although its performance is often more efficient than that of parallel or automatic backtrack search,[10] no procedural knowledge specific to the scene domain is used. What is needed are procedures, called methods,[11] attached to each model that can efficiently guide the search

process for instances of the model. These methods must be able to use a combination of data-driven and goal-driven techniques.

Figure 1. Input sketch for Mapsee2 shows the lower mainland of British Columbia.

Table 1.
A Geo-System schema instance.

TYPE:	
Class	Geo-System
Name	Geo-System-3
Lableset	{Landmass, Mainland}
Part-of	{World}
Composition	{River-System, Road-System, Town, Shore, Mountain-Range}
COMPONENTS:	
World	World-1
Road-Systems	{Road-System-1}
River-Systems	{River-System-1, River-System-2}
Shores	{Shore-2, Shore-8, Shore-9}
Towns	{Town-1, Town-2, Town-3, Town-4}
Mountain-Ranges	{Mtn-Range-1, Mtn-Range-2, Mtn-Range-3, Mtn-Range-4}
Chains	{C3, C4, C5, C6, C7, C29, C27, C19, C30, C32, C33, C31, C25, C26, C20, C21, C34, C35, C38, C39, C41, C42, C43, C24, C28, C22, C23, C36, C37, C40, C45, C17, C8, C9, C10, C11, C12}
Regions	{R1, R2, R3, R10, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R22, R23}

Limitation 7. A correct segmentation of the input image is necessary. Erroneous cues resulting from a poor segmentation will inevitably invoke inappropriate models leading to improper or empty scene interpretations. The problem can be ameliorated by a conservative initial segmentation designed to invoke only appropriate models. The resulting partial interpretations can then be used in a cycle of perception[4] to refine the parameters of the segmentation in a context-sensitive way. However, this approach appeals to a control mechanism, which is external to the basic methodology itself. Furthermore, for complex imagery, there may be no appropriate segmentation strategy that yields sufficient "correct" cues to drive the interpretation process. The disappointing performance of classification and region-growing algorithms for interpretation illustrates this phenomenon.

Of these seven shortcomings, the first four can be considered descriptive adequacy issues while the last three concern procedural adequacy.

Achieving descriptive adequacy

In response to the shortcomings discussed above, we have been exploring schemata as a suitable representation for knowledge.[9] Others have also advocated this representation.[11] Our experiments using schemata for visual perception have resulted in a program called Mapsee2. It automatically interprets hand-drawn sketch maps of cartographic scenes, producing a hierarchical structural description of the scene. Figure 1 is an input sketch map of the lower mainland in the Vancouver, British Columbia, area. It depicts a large body of water, the Strait of Georgia, on the left, the mainland on the right, and three islands in Howe Sound at upper left. On the mainland, the cities of Vancouver, North Vancouver, West Vancouver, and Surrey are represented by the "squiggly" lines. The "peaks" north of the cities are the North Shore Mountains. The cities are connected by roads, which cross the Fraser River at various points and cross Burrard Inlet at the Lions' Gate Bridge. (Some features of the Vancouver area have been stylized in this map to conform with the symbols understood by the system.)

The sketch map domain was chosen for the following reasons:

(1) Sketch maps capture in a simple form fundamental problems in representing and applying visual knowledge.
(2) Techniques for understanding maps have application in interpreting real imagery. In particular, sketch maps have been used to guide the cooperative interpretation of aerial photography.[12]
(3) By using the same task domain, the capabilities of schema-based systems can be compared directly with the well-understood properties of network consistency methodology.

The knowledge base used in Mapsee2 is a network of schema models. Each model represents a *class* of objects, providing a description of the generic properties of every

member of the class and specifying the possible relationships of the class with other schemata in the network. When a schema is used to represent a particular scene object, known or hypothesized to exist in a given sketch map, the class is used to generate a schema *instance*. For example, Table 1 shows an instance of the Geo-System class. This instance, named Geo-System-3, represents the Vancouver metropolitan area in the sketch map. The instance contains a number of defining properties, including a Labelset, indicating that the instance has been interpreted both as a Landmass and the Mainland; a set of relations with other schema classes; and a set of components, which are also schema instances.

Schemata represent complex scene interpretations as specific compositions of simpler schemata, forming a *composition hierarchy*. A complex scene object is recognized by recursively recognizing its component parts so that the internal constraints of its schema are satisfied. Figure 2 shows the composition hierarchy used in Mapsee2.

In this hierarchy, each node is a schema class and the arcs between nodes depict relations between schemata. Looking downward, the arcs represent *composition*, whereas in the upward direction they represent its inverse relation, Part-of. The intuitive interpretation of the hierarchy is that a cartographic World is composed of some number of geographic systems, called Geo-Systems, which are, in turn, composed of combinations of River-Systems, Road-Systems, Mountain-Ranges, Shorelines, and Towns. Each of these is, in turn, composed of simpler subschemata, finally terminating in the primitive input sketch lines, called *chains*, and the "empty space" *regions* bounded by the chains. Conversely, the hierarchy can be viewed as a *part-of hierarchy*, representing, for example, that Town schemata are component parts of both Geo-Systems and Road-Systems.

Schemata provide an important improvement in descriptive adequacy over network consistency and related representations. To substantiate this claim, in this section we examine how schemata overcome the first four of the seven objections outlined above.

Overcoming Limitation 1. The distinction between models and objects is unnecessary. Instead, schemata are models for scene objects at various levels of abstraction. The interpretation of a scene is expressed as a structural network of instantiated schema instances instead of being projected onto atomic labels for primitive objects. The interpretation is represented explicitly and need not be reconstructed from the labels. For example, Mapsee2 produces a network description of the lower mainland, which is shown as a color-coded image in Figure 3. The description consists of seven Geo-Systems: four are Islands, one is Sea, one is Lake, and the land area bordering the frame is interpreted as the Mainland. Mapsee2 discovers two separate Road-Systems, one of which is located on the Vancouver Mainland and contains the Roads, Bridges, and Towns in that area. The second Road-System is an isolated Town and Road on the

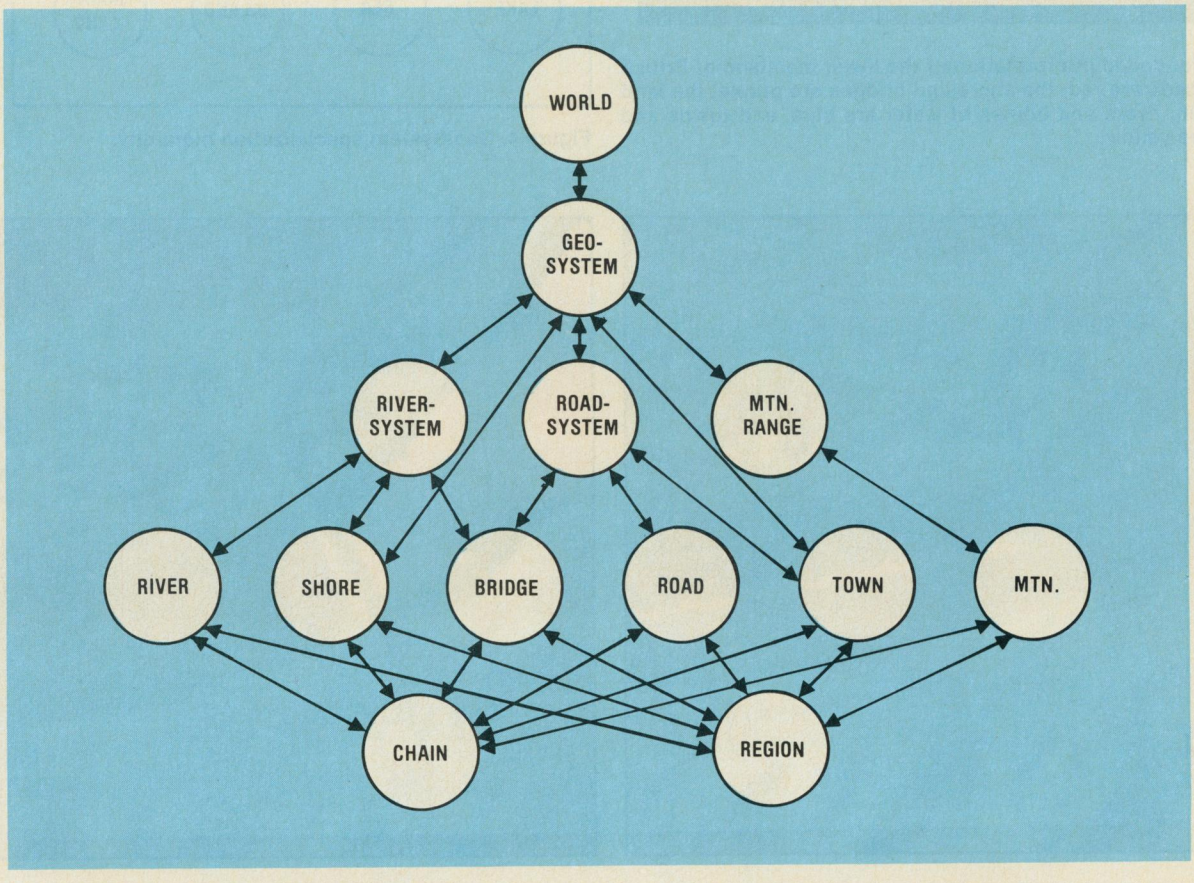


Figure 2. Mapsee2 composition hierarchy.

Sechelt Peninsula located in the upper left corner of the map. Finally, the Mainland has two River-Systems, one representing the Fraser River system and the other the First Narrows connection between the Sea and Burrard Inlet (which is interpreted as a Lake).

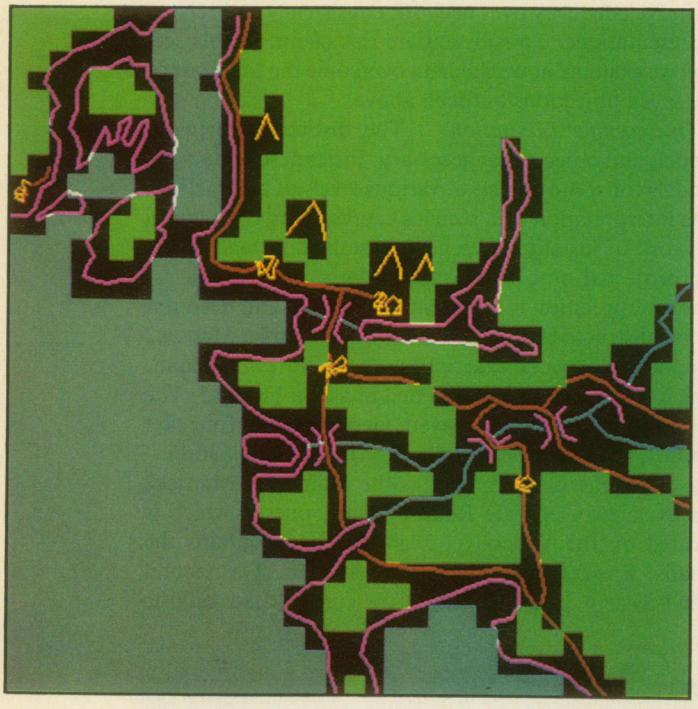


Figure 3. Color-coded Interpretation of the lower mainland of British Columbia. Roads are red, the shore and bridges are purple, the land mass is green, rivers and bodies of water are blue, and towns and mountains are yellow.

Overcoming Limitation 2. Schema models express scene relationships at an appropriate level of abstraction. A model constrains both the possible relationships of its components lower in the composition hierarchy and of the higher schemata of which it can be a part. Thus, constraints need not be localized to small neighborhoods of the image but can express global scene relationships in a natural way.

For example, in Figure 2, Road-Systems constrain their component parts to be connected Roads, Towns, and Bridges and simultaneously force the Geo-Systems, of which they are parts, to be Landmasses, as shown in the interpretation in Figure 3.

Overcoming Limitation 3. Schemata support an intensional representation for object label sets. There is no explicit representation of all possible final interpretations

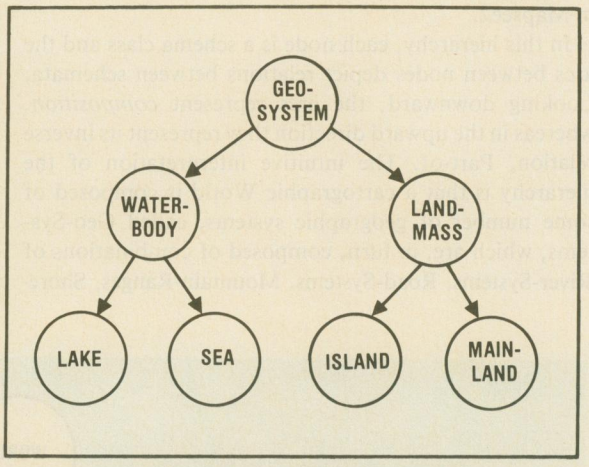


Figure 4. Geo-System specialization hierarchy.

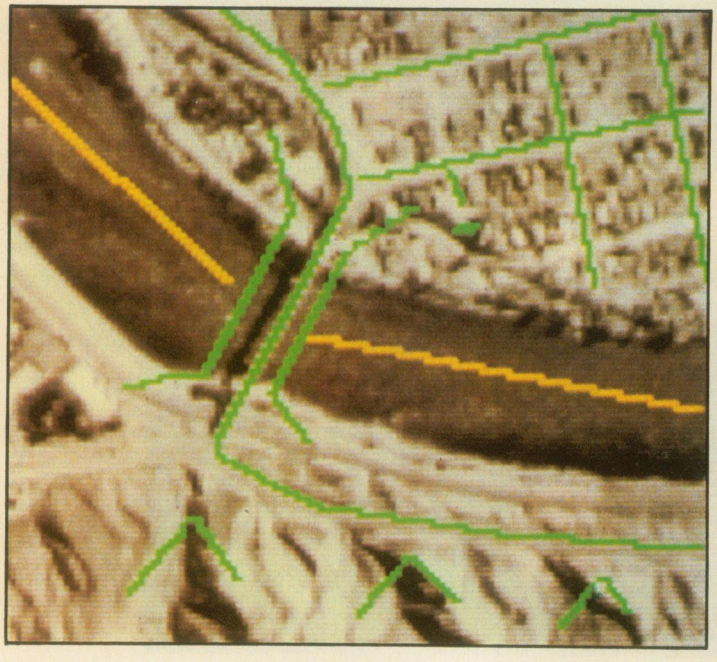


Figure 5. Sketch map superimposed on image of Ashcroft, B.C.

Figure 6. Missee River-1 interpretation.

for an object. Instead, a schema instance implicitly stands for all labelings that are consistent with its current description. The labels in this label set are not mutually exclusive but form a hierarchy called the specialization hierarchy. The top node of this hierarchy is a schema representing a general class of objects. Each offspring node in the hierarchy represents a specialization of the class of its parent.

For example, the specialization hierarchy for Geo-Systems is given in Figure 4. A Geo-System is initially a set of undifferentiated regions and embedded chains in a sketch map. As additional constraints on a Geo-System are found during recognition, its interpretation can be refined first to either a Landmass or a Waterbody and finally to one of Island, Mainland, Lake, or Sea. Each of these specializations can be a distinct type of schema in the hierarchy.

Overcoming Limitation 4. The size of the label set for each schema does not grow significantly with the complexity of the domain. Each schema model has only a small number of possible labels. As a result, no compiler is required to exhaustively consider and represent explicitly all legitimate relations among objects in any scene. Conceptually, the function of the compiler has been distributed among the schema models.

For example, in the Geo-System schema, the possible labels for any instance are only the nodes in its specialization hierarchy. As components are added to an instance, or the interpretation of any existing component is modified, the possible interpretations for the Geo-System are reexamined and possibly refined. This computation is local to the schema and need not be computed beforehand for all possible scenes.

Achieving procedural adequacy

Controlling search in artificial intelligence systems is not well understood. Most of the theoretical work in knowledge representation has focused on issues of descriptive adequacy. A prime motivation for the interest in schemata is their ability to represent both descriptive and procedural knowledge in a natural and effective manner, thereby helping to overcome the three procedural adequacy limitations.

Overcoming Limitation 5. Schemata can support a hierarchy of cues and models. By representing complex objects as compositions of simpler components, a search for these objects can exploit the structure of the composition and specialization hierarchies. Schema at the bottom of the composition hierarchy are invoked, as before, by context-free cues derived directly from the image features. In Mapsee2, the low-level cues are configurations of the chains present in the input sketch. Schemata higher in the composition hierarchy are invoked by abstract cues. When an instance has been fully instantiated (or nearly so), it can be used as a high-level cue to invoke schemata directly above it (bottom-up search) or schemata directly below it (top-down search) in the composition hierarchy. By using this cue/model hierarchy, the disparity between low-level cues and high-level models is avoided.

Overcoming Limitation 6. Procedural knowledge can be used to guide search efficiently. Three distinct modes of search are possible: top-down, bottom-up, and a hybrid mode that combines desirable aspects of both.

In top-down search, a schema is proposed as a likely subgoal by some schema higher in the composition hierarchy. Eventually, the subgoal must either succeed or fail, returning control to its caller. Unfortuantely, a commitment must be made to the subgoal before any of the schema's expertise becomes available to help guide the search. Furthermore, the exploration of possible alternative subgoals is completely failure driven. Consequently, top-down techniques are best viewed as appropriate for confirming the last details of an instance after it has been established as a likely hypothesis.

At the other extreme, bottom-up search is warranted if few or none of the componets of an instance have been found. The components of the schema are used as cues to invoke its methods. Once invoked, a method checks the consistency of the new component with the internal description of the instance. If they are compatible, the schema's description is updated and its label set is possibly refined. Bottom-up search permits multiple active hypotheses. However, no particular schema is in control to guide the recognition process.

The hybrid mode provides a mechanism that allows top-down search to give overall guidance, yet permits bottom-up techniques to circumvent the inefficiencies of purely top-down techniques. In bottom-up search, when a schema has successfully incorporated a new component into its description, its method suspends awaiting the recognition of additional components. Instead, the method can retain control to direct the search for those components. For example, the method can focus the attention of the segmentation procedure to those areas of the image where its schema's components are likely to be found. If the method is successful, then another cue will be discovered in the scene that matches its schema. On the other hand, if the method fails or finds components that act as cues for other schemata, then the methods of those instances will be invoked instead. The advantage of this technique is that the schema can employ its methods to guide the search for its components without a commitment to top-down search. As long as it is successful, the schema retains control. However, as soon as components are found that can be part of a different schema, control is appropriately transferred to that schema.

Overcoming Limitation 7. High-level knowledge can guide segmentation processes. Glicksman[12] has shown that cooperative interpretation using schema-based systems can integrate information from separate information sources. His Missee system uses as input an aerial image and a sketch map drawn on top of the image outlining the major geographical objects that can be found. For example, Figure 5 is an image of part of Ashcroft, British Columbia, with an overlayed map of a river (shown in yellow), a mountain range, a bridge, and a road system (all shown in green). Mapsee2 is first used to provide a structural description of the map. This interpretation is then used to guide the spectral segmentation process operating on the aerial image. Figure 6 shows the

final segmentation for River-1 in the image (again shown in yellow). Corresponding image regions are also found for the other objects represented in the sketch map description.

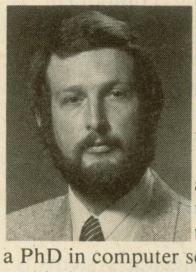
We have argued that visual perception requires knowledge of the objects of interest to the system. This knowledge necessarily includes both descriptive and procedural aspects. Furthermore, efficient visual processing requires a search of the knowledge base with a combination of data-driven and goal-driven methods. From the perspective of the knowledge representation used, network consistency—although successful—has a number of limitations. Schema representations offer a solution to these difficulties. ■

Acknowledgments

We are grateful to Jan Mulder, Rachel Gelbart, and Jay Glicksman for their contributions to Mapsee2. This work was supported by NSERC under grants A9281, A5502, and SMI-51, the University of British Columbia, and NSF grant MCS-8004882.

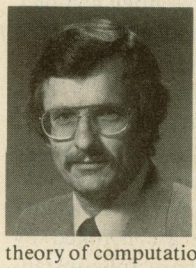
References

1. D. Marr, *Vision*, W. H. Freeman, San Francisco, 1982.
2. H. G. Barrow and J. M. Tenebaum, "Recovering Intrinsic Scene Characteristics from Images," *Computer Vision Systems,* A.R. Hanson and E.M. Riseman, eds., Academic Press, New York, pp. 3-26.
3. L. G. Roberts, "Machine Perception of Three-Dimensional Objects," *Optical and Electro-Optical Information Processing*, J. T. Tippet et al., eds., MIT Press, Cambridge, Mass., 1965, pp. 159-197.
4. A. K. Mackworth, "Vision Research Strategy: Black Magic, Metaphors, Mechanisms, Miniworlds, and Maps," *Computer Vision Systems*, A.R. Hanson and E. M. Riseman, eds., Academic Press, New York, 1978, pp. 53-59.
5. A. K. Mackworth, "How to See a Simple World," *Machine Intelligence,* Vol. 8, E.W. Elcock and D. Michie, eds., Halstead Press, New York, 1977, pp. 510-537.
6. D. L. Waltz, "Understanding Line Drawings of Scenes with Shadows," *The Psychology of Computer Vision*, P. H. Winston, ed., McGraw-Hill, New York, 1972, pp.19-91.
7. A. K. Mackworth, "Consistency in Networks of Relations," *Artificial Intelligence,* Vol.8, No. 1, 1977, pp. 99-118.
8. A. K. Mackworth, "On Reading Sketch Maps," *Proc. Int'l Joint Conf. Artificial Intelligence,* Aug. 1977, pp. 598-606.
9. W. S. Havens, "Recognition Mechanisms for Hierarchical Schema-Based Knowledge Representations," *Int'l J. Computers and Mathematics*, Vol. 9, No. 1, Pergamon Press, 1983, pp. 185-199.
10. A. K. Mackworth and E. C. Freuder, "The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems," tech. report 82-6, Computer Science Dept., University of British Columbia, Vancouver, B.C., 1982.
11. T. Winograd, "Frame Representations and the Procedural-Declarative Controversy," *Representation and Understanding*, D. G. Bobrow and A. Collins, eds., Academic Press, New York, 1975, pp. 185-210.
12. J. Glicksman, "Using Multiple Information Sources in a Computational Vision System," *Proc. Int'l Joint Conf. Artificial Intelligence*, Aug. 1983, pp. 1078-1080.



William Havens is an assistant professor of computer science at the University of British Columbia. His research interests focus on the representation of knowledge for computational vision and its applications in remote sensing and robotics.

Havens received a BSc degree in 1969 and an MSc degree in 1973, both in electrical engineering, from Virginia Polytechnic Institute and State University and a PhD in computer science from the University of British Columbia in 1978. He held positions at Simon Fraser University and the University of Wisconsin at Madison before returning to UBC in 1981. He is a member of IEEE, ACM, and Phi Kappa Phi.



Alan Mackworth is an associate professor of computer science at the University of British Columbia and director of the UBC Laboratory for Computational Vision. He received a bachelor's degree from the University of Toronto in 1966, a master's from Harvard University in 1967, and a doctorate from Sussex University in 1974 before moving to UBC in the same year. Since then his research has focused on the theory of computational vision and applications in remote sensing and geographical sketch maps. He has worked on the representation and use of knowledge in vision systems, including surface orientation representations, network consistency constraint satisfaction algorithms, and schema-based systems.

The authors' address is Dept. of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada V6T 1W5.