Knowledge-Directed Vision: Control, Learning, and Integration

Bruce A. Draper[1], Allen R. Hanson, Edward M. Riseman[2]

Affiliations

Bruce A. Draper
Assistant Professor
Computer Science Department
Colorado State University
Fort Collins, CO 80523
Home: (970) 206-0642
FAX: (970) 491-2466
draper@cs.colostate.edu

Allen R. Hanson
Professor
Computer Science Department
University of Massachusetts
Amherst, MA 01003
Office: (413) 545-2746
FAX: (413) 545-1249
hanson@cs.umass.edu

Edward M. Riseman
Professor
Computer Science Department
University of Massachusetts
Amherst, MA 01003
Office: (413) 545-2746
FAX: (413) 545-1249
riseman@cs.umass.edu

---

[1] Computer Science Department, Colorado State University, Fort Collins, CO 80523.

[2] Computer Science Department, University of Massachusetts, Amherst, MA 01003

Abstract[3]

The knowledge-directed approach to image interpretation, popular in the 1980's, sought to identify objects in unconstrained two-dimensional images and to determine the three-dimensional relationships between these objects and the camera by applying large amounts of object- and domain-specific knowledge to the interpretation problem. Among the primary issues faced by these systems were variations among instances of an object class and differences in how object classes were defined in terms of shape, color, function, texture, size, and/or substructures.

This paper argues that knowledge-directed vision systems typically failed for two reasons. The first is that the low- and mid-level vision procedures that were relied upon to perform the basic tasks of vision were too immature at the time to support the ambitious interpretation goals of these systems. This problem, we conjecture, has been largely solved by recent advances in the field of 3D computer vision, particularly in stereo and shape reconstruction from multiple views. The other impediment was that the control problem for vision procedures was never properly addressed as an independent problem. This paper reviews the issues confronted by knowledge-directed vision systems, and concludes that inadequate vision procedures and the lack of a control formalism blocked their further development. We then briefly introduce several new projects which, although still in the early stage of development, are addressing the complex control issues that continue to obstruct the development of robust knowledge-directed vision systems.

1. Introduction

The goal of image understanding systems often involves the identification of objects in two-dimensional images, and the establishment of three-dimensional relationships among

[3] This work was supported in part by the Advanced Research Projects Agency through contracts DACA76-92-C-0041 and F30602-91-C-0037, by the National Science Foundation through grants MCS7918209 and CDA-8922572, and by the Army Research Office through grant DAAH04-95-1-0068.

the objects and between the objects and the viewer. This transformation of signals (the image) to symbols (the interpretation) in the visual domain is almost certainly the most complex sensory interpretation problem that exists for both human and machine. The vision problem is ill-defined, requires immense computation, and must operate robustly in widely varying contexts and under varying illumination. Yet perception is performed by humans in an immediate, effortless, and to a great extent subconscious manner. In contrast, it has turned out to be exceedingly difficult to build robust, autonomous computer vision systems other than for highly restricted domains.

The inherent difficulty of the problem stems from the nature of both the signals and the symbols. The image is a record of a spatially sampled discrete approximation to the scene luminance, which varies as a function of the incident illumination and viewpoint. With respect to object and scene recognition, this signal is further corrupted and degraded by occlusions, specular reflections, interreflections, atmospheric conditions, lens distortions, the normal foreshortening effects of perspective projection, and digitization. In unconstrained domains it is impossible to develop an effective analytical model that inverts or compensates for all of these effects.

The symbols, for their part, are equally complex. Our goal is to interpret images in terms of object classes, both natural and man-made, and emergent scene properties. This requires recognizing instances of classes such as “house” and “tree”, even though particular instances of these classes differ quite dramatically from each other and from the class prototype. It also requires recognizing many different types of objects, even though some object classes are defined in terms of shape, while others are defined in terms of color, texture, function, context, size, substructures, or a combination of these factors. To identify instances of a class, computer vision systems must first discover or have specified the subset of possible attributes that members of the class share, and the acceptable range of variation for these attributes.

To address these issues, many researchers in the 1980's adopted the knowledge-directed vision paradigm, in which large amounts of information were collected about each object class, including what characteristics (shape, color, etc.) its members shared, and how much variation was allowed for each characteristic within the class. The aim was to build intelligent systems that could match these complex class descriptions to image data through a library of reusable vision procedures (sometime cast as knowledge sources or production rules, according to the AI technology of the 1980's).

Unfortunately, the available vision procedures were fairly impoverished relative to the lofty goals of these systems. Much of the focus was diverted into developing low and intermediate-level vision algorithms or to operating in domains where rigid 3D shape models were available and sufficient. This eventually resulted in a natural development of 10-20 (or more) subfields, each focusing on a single subtask. Within these subfields, theories were developed and tested, and different solution methodologies adopted. As a result, there are now several good and improving algorithms for many subdomains, including edge and line extraction (straight and curved), stereo analysis, tracking over time, depth from motion (two-frame and multi-frame), shape recovery, CAD/CAM model matching, 3D pose determination, and color-based focus of attention, to name just a few.

Of particular interest are the advances in the extraction of depth and shape information from images. New techniques in multi-frame shape reconstruction [Tomasi and Kanade 92] and in stereo [Matthies 92; Kanade and Okutami 94; Schultz 94], as well as the development of direct 3D sensors such as LADAR, IFSAR and light-stripe sensors, allow us to measure 3D features and to adjust for the effects of perspective distortion as never before. Because of these and other advances, we believe it is time to reconsider computer vision “in the large”, and to work toward systems that can interpret natural scenes, with all the complexity, variations and counter-examples they invariably contain, even if we still have to limit ourselves to partially constrained domains in the near-term.

As part of this reassessment, this paper reviews the issues at the core of knowledge-based vision, emphasizing those insights and intuitions that may be unappreciated today. At the same time, we turn a critical eye toward this approach, discussing the factors (beside the relative immaturity of the vision procedures) that we believe led all too often to failure, or at best to partial success. Note that there is no attempt to survey the many systems that were built; that task has already been accomplished in [Binford 1982; Brown 1987; Jolion 1994; Wallace 1988]. Instead, the discussion will be centered around issues. In some cases, particularly with regard to failures, we will focus on the VISIONS/Schema System, a system for interpreting outdoor natural scenes which was first reported on in 1978 [Hanson and Riseman 78] and subsequently refined for eleven years [Draper, at al 89]. We hope the reader will understand that by focusing on our own work, we can draw upon our personal experience and freely criticize its shortcomings. The point of the discussion, however, is to summarize our understanding of knowledge-directed vision systems in general, based not only on the Schema System but also the reported work on SIGMA [Hwang et al 86], SPAM [McKeown et al 85], PSEIKI [Andress and Kak 88] and other systems.

In particular, we argue that while these systems addressed (and to some extent solved) many critical issues, their success was limited not only by the relative immaturity of the field, but also by fundamental and still-open problems in control. Section 2 of this paper discusses the issues addressed in knowledge-directed vision that we believe are still relevant, while Section 3 bluntly describes some of the problems they faced. Section 4 and then introduces new lines of research which, although still in the early stages of their development, we believe will lead to solutions to the control problems discussed in Section 3. Section 5 concludes with our vision of the future of knowledge-based image interpretation.

It should be noted that this paper focuses on systems that produce internal symbolic interpretations of static scenes, with very little discussion of image sequences and temporal dependencies. This is not because dynamic motion cues are not useful visual information, but rather because this research area is complementary to the issues and capabilities that we will be examining. Clearly, people can easily interpret still photographs, and do so all the time. Therefore the interpretation of general scenes should not require sequences, multiple views, or any other data that allow 3D depth information to be directly computed (in contrast to indirect inference based on assumptions about the scene). However, it is also true that motion analysis can significantly aid in 3D scene interpretation by segmenting independently moving objects, providing depth cues, supplying new views of objects, etc. Nevertheless, all of the issues discussed here, and the mechanisms developed to deal with them, are necessary for dynamic scene interpretation as well.

Many researchers have had success with systems that produce actions directly in response to streams of images (e.g. Pomerleau 1993). However, many of these systems do not produce detailed internal symbolic representations of the environment, which we believe are necessary for many types of intelligent behavior. For an interesting discussion on these issues, see Tarr and Black (1994).

2. Knowledge-Based Image Understanding

Simply put, interpreting an image is a matter of establishing correspondences between the image (i.e. the signal) and objects in the knowledge base (the symbols). Thus, it is not sufficient to reconstruct only the geometry of a scene, for example by describing orientations and positions of surfaces. A semantic interpretation must also include the types (or classes) of objects present in the scene being imaged, their spatial relationships to each other and the viewer, their semantic interrelationships (such as part/whole relations), and perhaps their functional properties when relevant.

2.1 Object Recognition and Intra-class Variation

Identifying instances of general object classes is an extremely difficult problem, in part because of the variations among instances of many common object classes, many of which do not afford precise definitions. Houses, for example, come in many different shapes and sizes (not to mention colors and textures). Object recognition is therefore more than recognizing fixed shapes, and indeed is more than strict matching on any single characteristic. Object recognition requires flexible matching on a variety of characteristics, with the ability to handle exceptions and deviations. For example, houses can usually be recognized based on shape constraints (closed volumes with vertical walls, mostly at right angles), commonly found substructures (e.g. windows, doors, chimneys) and a limited pattern of well-delineated colors (generally not more than two). Other objects, such as trees, present quite a different array of features and problems. Object recognition therefore requires a flexible and multi-faceted matching scheme. Most simple matching algorithms are simply not adequate by themselves, although they may be part of the solution in that they may work well for certain objects or within specific domains.

It should be noted that, in some cases, it is possible to reduce intra-class variability by subdividing a class until the members of the subclasses share a property, such as a single fixed (or perhaps parameterized) shape. In such cases, straightforward matching strategies can be quite effective. Unfortunately, this is not a universal solution; the class of houses, for example, would have to be subdivided into many types, perhaps to the point of individual homes. At that point, it is not clear which is the greater problem - intra-class variability or the proliferation of subclasses!

Approximate and flexible matching in turn requires a different kind of object model. Object class models must now specify those characteristics that are common across the elements of the class, and to what extent they vary. Object models should also include characteristics or features which are shared by many but not all of the members of the class, since the presence of each such feature can increase the system's confidence (or probability) of a match for many object instances. In general, we refer to this type of information about an object as *object knowledge* rather than object models, both for historical reasons and because the latter term generally refers to rigid or parameterized shape descriptions.

(a)

(b)

Figure 1. An image of a typical urban setting. (a) The two houses show the intra-class variability in architectural styles, colors, shapes, and other geometric features for a common object. (b) A close-up view of the left house in the area around the right lower corner of the dormer window illustrates the futility of attempting to recognize objects directly from the pixel data.

2.2 Inter-class Variability and the Knowledge-Directed Approach

If the aim of a recognition system is to identify instances of just one object class, then flexible matching strategies are not actually necessary. A house recognition system (to continue with the example) could be built that always searched for large rectilinear structures, which it then verified by looking for key subparts, such as doors and windows, and by counting the number of distinct colors. Such a system could be reasonably effective at identifying houses. Unfortunately, it would be completely incapable of recognizing trees or roads.

The problem is that classes differ from each other not only in the values of their features, but in terms of what features are even defined. It would be difficult and probably meaningless to count the number of distinct colors (mostly greens and browns) in a tree, even though such a color count is well defined for houses. Yet the range of colors for a given type of tree is quite distinctive (except possibly in autumn). Combined with the micro- and macro-texture associated with leaves, branches and crowns, the color of a tree's leaves produces a characteristic color variation that is probably the best single feature description for tree recognition. Thus, both class definitions (i.e. houses and trees) include color features, but how those features are represented and matched is obviously quite different. We therefore once again emphasize the fundamental conclusion that not all object classes are defined in terms of the same attributes, and these attributes may be used in various ways within the matching or interpretation process.

One response among researchers to this problem is to search for a single all-inclusive representation and matching scheme, capable of distinguishing instances of any class of objects from any other. One entry in this category, for example, are the eigenvector matching approaches (e.g. [Nene, et al 94]) which, although they have not yet shown the ability to recognize all object classes, have had some success at partially viewpoint invariant recognition of non-trivial objects. Alternatively, the knowledge-directed approach seeks to take the union of all special-purpose techniques. Color, for example, can be represented and matched as a range of hues, as a color histogram, and/or as a count of distinct colors (among other possibilities). Possible shape representations include CAD models, fractal models, and surface splines. Object models (i.e. object knowledge) contain whatever characteristics and features are defined for the object class (including possibly appearance models, such as eigenvector representations), as long as there is a known procedure to detect or measure them. Customized *control knowledge* must be used to select which features to look for first, and to combine the results of multiple partial matches.

2.3 Structuring Knowledge

In principle, the knowledge-directed approach to image interpretation is an elegant solution to the problems of inter-class and intra-class variation. Every object class is defined in terms of whatever features are (relatively) constant among its members, and every feature has a procedure (or production rule, or knowledge source) for measuring it. Control knowledge associated with each object class determines which features to measure and in what order to measure them.

Unfortunately, how to collect and structure this knowledge is a major open issue. Knowledge collection (including knowledge engineering) will be discussed later; most research in the 1980's was more concerned with the issue of how knowledge is structured. The VISIONS system, for example, was one of several projects that began as traditional AI-style blackboard systems [Hanson and Riseman 78]; see [Andress and Kak 88; Hwang et al 86] for examples of other blackboard-based vision systems. Knowledge sources represented specific knowledge about specific objects, for example how to match the color of a tree. As the VISIONS/Schema System evolved, successively more structure was added, mostly to allow more and more specific matching strategies without creating interpretation interference problems[4] [Draper et al 89]. In particular, the VISIONS/Schema System introduced the notion of a *schema*, which was an active process that encapsulated the knowledge about an object class and had its own, private memory and explicit, procedural control strategies. In the current terminology of AI, the VISIONS/Schema System evolved from a blackboard architecture into an active agents paradigm.

Other researchers took different approaches to structuring knowledge. The expert system paradigm in which knowledge is encapsulated in production rules was also popular (e.g. [Ohta 80; McKeown et al 85]). Unfortunately, this approach made it difficult to prevent new knowledge from interfering with old, and also required special (some would say ad-hoc) mechanisms for introducing control knowledge. SPAM, for example, divided its production rules into five processing phases in order to interpret aerial images of airports [McKeown et al 85]. Frame-based systems [Minsky 75] approached knowledge at a more macroscopic level, as each frame collected information into larger chunks to be invoked as a unit. This allowed control to be customized in ways that were not possible in simple rule-based systems, as demonstrated by 3DFORM [Walker, et all 88].

Many of the details of how knowledge was structured in these systems is arguably no longer relevant. The VISION/Schema System, for example, would be designed differently

---

[4] Interpretation interference is when new knowledge about an object class or knowledge about a new object class interferes with the recognition of other classes. It can occur in the traditional blackboard framework because all data is public and can trigger any knowledge source

today given the existence of object-oriented programming languages and active agent systems. However, the notion of using flexible object knowledge, including control knowledge, to combat inter-class and intra-class variation, and of encapsulating this knowledge according to object class, remain germane to the construction of object recognition systems today.

2.4 Accessing the Knowledge Base: Indexing

The same broad nature of object knowledge that makes it possible to describe large classes of objects also makes it difficult to match them. After all, if an object model permits enough variations, many features may have to be missing before a system can reject the hypothesis that the object is present. The task of selecting which objects to match against the data, sometimes called the object indexing problem (or just the indexing problem [Rosenfeld 1984]), is therefore critical.

Unfortunately, none of the approaches to object indexing developed thus far are completely satisfying. Most knowledge-directed systems worked in a limited domain, where they could always look for a fixed set of objects, such as houses and roads in aerial images of suburban scenes [Hwang et al 86] or runways, taxiways, tarmacs and terminals in aerial images of airports [McKeown et al 85]. The VISIONS/Schema System was somewhat more sophisticated in that it exploited contextual indexing, for example by triggering a schema to look for telephone poles anytime a road was recognized, on the basis that telephone poles often run alongside roads. The VISIONS/Schema System also had contextual objects, such as “road scene” and “house scene”, which were used to group contextually related objects. More recently, the CONDOR system expanded the idea of contextual processing to include maps as contextual clues when searching for objects [Strat 91].

Contextual indexing is a powerful technique that we believe is clearly part of the solution to the indexing problem. It does not address, however, people's ability to interpret photographs out of context. Other researchers have introduced techniques for indexing into databases based on specific features or attributes (e.g. [Burns and Riseman 92]). Recently, Nayar has introduced an algorithm for indexing into a database of 3D shapes directly from a (2D) image, and techniques such as this are again likely to be part of the solution to the indexing problem. These techniques are not general-purpose, however, in that they are restricted to indexing into databases of rigid shapes. It is unclear whether most general object classes, such as houses or trees, can be indexed (or recognized) based solely on these techniques.

2.5 Multiple levels of representation.

There is a large representational gap between the initial sensory data (pixels and their properties) and the high-level symbolic descriptions that constitute an interpretation. Large inferential jumps, for example from pixels to much more abstract representations such as surface and volume entities, or object identities, are extremely errorful and therefore not widely used except under very constrained conditions.

Rather, the idea of hierarchical representations of knowledge, in which the levels correspond roughly to the vocabulary of the intermediate representation of an interpretation, has gained wide acceptance among vision researchers [Barrow and Tenenbaum 1978; Hanson and Riseman 1978b; Hanson and Riseman 1987b; Marr 1982; Rosenfeld 1984]. Since an interpretation can be viewed as a correspondence between image features and knowledge classes, it is clear that the descriptive vocabulary for images must be reflected in the hierarchical representation of the knowledge base. That is, the system must be able to establish the correspondences based on extractable and derivable image features, and these same image features must form the basic descriptions of the objects and object classes in the knowledge base. Viewed in this manner, it is clear why many of the knowledge representations currently in use encode knowledge of objects (individual instances, classes, and descriptions) and of events (actions, situations, cause and effect) as a combination of data structures and interpretation procedures.

2.6 Uncertainty and Ambiguity

There is a significant amount of inherent ambiguity in the interpretation process. In order arrive at an unambiguous final result, a knowledge base must include a sufficiently rich set of constraints and flexible mechanisms for manipulating uncertain hypotheses until there is a convergence of evidence.

This argues for a methodical construction of an interpretation across increasingly more abstract descriptions of the image. The matching processes responsible for generating correspondences between images and object classes will inevitably make errors and may generate multiple competing hypotheses to describe portions of the scene. This results in uncertainty as to the “correct” interpretation at all levels of representation. In order to use these uncertain hypotheses, their confidence level must be taken into account. Treating the hypotheses as evidence for (or against) a set of models and the confidence values as belief in the evidence leads to the general idea of inference mechanisms operating over the knowledge base.

The inference mechanisms must be capable of pooling or combining evidence from multiple sources in a consistent manner, and propagating this pooled information over the knowledge base, subject to any relevant constraints or relationships. The inference mechanisms support the establishment of image-to-knowledge correspondences by providing a principled mechanism for building partial interpretations at various levels of representation. Many different forms of reasoning under uncertainty have been used in computer vision systems (including Bayesian inference, Dempster-Shafer evidential reasoning [Wesley 86; Andress and Kak 88], and fuzzy reasoning [Sanchez and Gupta 1984; Zimmermann 1985] with varying degrees of success. (A review of inference mechanisms is beyond the scope of this paper; for more details see [Shafer and Pearl 1990]). The PSEIKI system in particular used Dempster-Shafer evidential reasoning within the context of a blackboard-based image interpretation system to resolve conflicting hypotheses [Andress and Kak 88], while recent work by Rimey and Brown [94] and by Binford et al [89] has focused on image interpretation using Bayesian nets.

Interestingly, there is anecdotal evidence that the exact form of evidential reasoning system may not be that important, so long as it can pool information from enough sources. The VISIONS/Schema System, for example, took a particularly simple view of evidence representation and combination. Confidence values were coarsely quantized on a five point ordinal scale, ranging from `no evidence' to `slim-evidence', `partial-support', `belief', and finally `strong-belief'. When combining evidence, heuristic and object-specific evidence combination functions were used to map combinations of evidence to confidence levels. These functions were typically written by specifying key pieces of evidence that allowed an object instance to be hypothesized and which could be used to compute an initial confidence level. Subsets of secondary confirming or discrediting evidence were used to raise or lower these confidences as such evidence was acquired. Specifications of these subsets, and the effect their confidence had on the overall confidence, was part of the knowledge engineering effort involved in constructing a schema.

This method of evidence representation and accumulation lacked considerably from a theoretical point of view, but it worked surprisingly well in interpretation experiments on images of New England house and road scenes [Draper, et al 89]. We suggest that while the ability to combine evidence is important, there is enough redundancy in many overconstrained interpretation tasks that systems can succeed without making optimal evidence combination decisions (although good evidence combination mechanisms are obviously preferable to bad ones). For a discussion on the relative importance of the structure of conditional dependencies versus the precise form of evidence propagation, see Pearl [1988]

3 Open Problems in Knowledge-Based Vision

The knowledge-directed vision research of the 1970s and 1980s recognized the complex nature of real-world object classes (as opposed, for example, to the assumption that the world is composed of rigid shapes). The major research emphasis at the time was on the knowledge required (both the appropriate structure and type), the control mechanisms, the representation of evidence, and the evidential inference mechanisms. This work led to several special-purpose systems, operating in strictly limited domains, which were built by exploiting knowledge of domain constraints (e.g. see [Ballard, Brown et al. 1978; Brooks 1981; Hanson and Riseman 1978b; Hanson and Riseman 1987a,b; Herman and Kanade 1986; Matsuyama 1989; McKeown, et al. 1985]). In retrospect, the relative success of these knowledge-based image understanding systems can be traced to a small world assumption, where the number of objects in the domain are few, the constraints on their descriptions are tight, and a complete world model is at least a possibility. Consequently, special-purpose systems were able to define, structure, and apply relevant task knowledge effectively. However, as the scope of a system broadened towards domain-independent, general-purpose applications, an unfortunate chain of events occurred: the size of the knowledge base increased, constraints on the object descriptions became looser to account for wider variability, fewer assumptions could be made about the types of image descriptions necessary for matching, and the complexity of knowledge and data increased substantially.

As stated earlier, many of the problems encountered by researchers as they attempted to broaden their system's domains were the result of immature algorithms. Edge extraction algorithms based on zero-crossings, for example, were just beginning to emerge in the late 1970s [Marr and Hildreth 80], and did not become truly robust until the mid-eighties. Even more dramatic changes have occurred in other areas; for example, in the early 1980's there were very few algorithms for reliably extracting any kind of 3D shape or depth information from images (with early work on stereo vision being a possible exception). Now there are several good motion algorithms for recovering the relative depths of points in space from point correspondences over two or more frames [Faugeras, et al 87; Tomasi and Kanade 92; Oliensis 94;], as well as algorithms for recovering the relative (3D) position and orientation of a camera from a set of image-to-model (2D-to-3D) point correspondences [Kumar and Hanson 89; Haralick, et al 89]. Add to this recent improvements in recovering depth through stereo processing of both traditional small-baseline, synchronous image pairs [Matthies 92; Kanade and Okutami 94] and larger-baseline, temporally distinct pairs [Schultz 94], the introduction of depth-from-focus techniques [Grossman 87; Krotkov 87], and the emergence of 3D sensors (e.g. LADAR, IFSAR, and structured light sensors), and it becomes possible to extract far more geometric information than was previously possible. In the 1980's, however, these techniques were not available, and as a result when geometric and/or other constraints were loosened on early knowledge-directed systems, their performance degraded badly.

We argue, however, that inadequate vision procedures were only one of the reasons these systems failed to generalize. As the scope of the problem domains broadened, the knowledge required --particularly the control knowledge -- became increasingly complex and difficult to apply. The knowledge engineering paradigm used to collect knowledge for small systems was inadequate for gathering the larger amounts of knowledge needed for more general systems. In addition, as the size of the knowledge base grew, the systems integration problems became more and more daunting. This was particularly true with regard to vision procedures; as the number and complexity of the component algorithms grew, the ability to organize procedures and to pass intermediate data from one procedure to the next in an efficient manner became increasingly critical.

3.1 Knowledge Engineering

One of the main impediments to wide scale experimentation with the Schema System was the manual labor required to design schemas. The knowledge base used for 2D road interpretation in [Draper, et al. 1989] took one and a half man-months to build[5] and contained about twenty objects. The roadline schema strategy shown in Figure 2 was only a small part of the control knowledge used in the Schema System, but it gives a good sense of the granularity at which control knowledge was specified. Another problem was that even when a knowledge base was finished, there was no guarantee that the interpretation strategies embedded in it were optimal or even adequate.

In general, knowledge base construction could be viewed as an exercise in experimental engineering, in which prototype schemas (or knowledge sources or rule sets) were developed using existing system resources. These schemas were then tested on a representative set of objects and images, failures were noted, and the schemas re-engineered to account for the failures. In particular, schemas were assembled by specifying (1) the appropriate set of vision procedures to be used, (2) control strategies to conditionally sequence their application, and (3) a function to map combinations of evidence into global confidence values. Failures could result from errors in any of these three areas, and while errors in evidence combination were (comparatively) easy to fix, errors in control strategies required debugging complex code modules. Worse still, some

---

[5] Other researchers have generally not reported the time it took to develop their knowledge bases.

failures were the result of inadequate or immature vision procedures. In such cases, schema development was interrupted while an improved vision procedure was constructed, a process that often involved a major research effort in its own right.

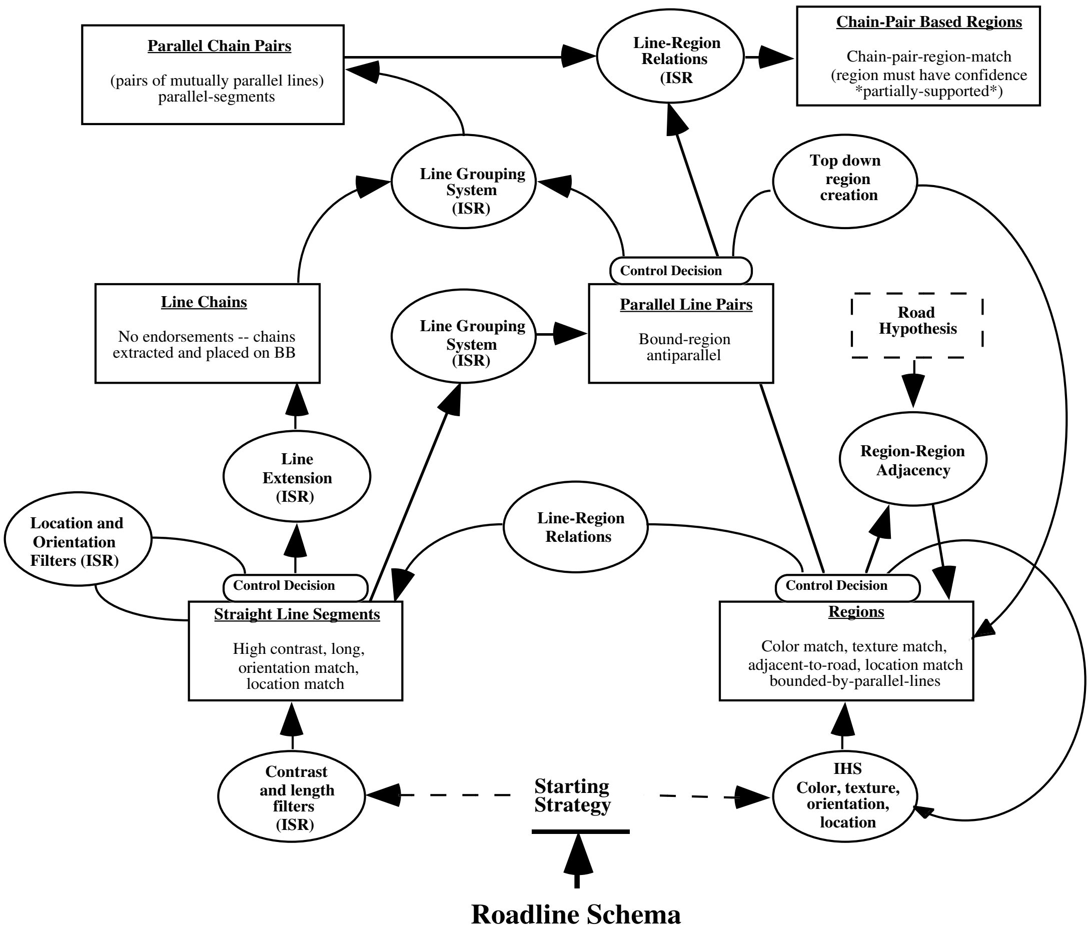

Figure 2. An example interpretation strategy for finding roadlines (the yellow or white lane control markings found on many hard-surface roads). This particular schema has two control threads, one based on image lines and the other on image regions.

3.2 Vision Procedures

Many types of vision procedures were developed by many researchers. The VISIONS/Schema System, for example, made extensive use of color and texture features, whereas systems designed for black-and-white aerial images (e.g. SIGMA, SPAM) tended to make more use of (2D) shape features. In addition, most of these systems tried to take advantage of contextual knowledge; SPAM, in particular, used detailed information about the functional relations between the parts of an airport (terminals, tarmacs, runways, etc.) to spatially constrain its search for specific airport components.

New procedures were almost always developed in response to observed failures, however, and in our experience these failures most often resulted from intra-class variations or changes in viewpoint. Intra-class variation would lead to failures when new instances of an object class were encountered that did not fit the previous model. The VISIONS/Schema System, for example, modeled houses as having windows and windows as having shutters. The first time a picture of a house without shutters was encountered, our knowledge about houses had to be loosened, leading to the unfortunate possibility of more false matches.

The other major source of failures during the development of the Schema System were changes in viewpoint and/or object orientation. Obviously, under the laws of perspective projection, a window appears as a rectangle when viewed head-on, but as a trapezoid if the camera position varies along one dimension and as an arbitrary quadrilateral if the camera position is unconstrained. As a result, the more orientations an object could assume, the looser the constraints on its (2D) shape model.

Much of our optimism about the future of knowledge-directed systems is based on the recent development of viewpoint-independent, 3D visual procedures. These advances actually began with the SCERPO system, which was capable of matching rigid 3D shape models to line segments extracted from images, no matter what the orientation of the target object [Lowe 85]; SCERPO also returned the 3D position and orientation of the object, relative to the camera. Since then, other techniques for matching 3D shapes to 2D images have been introduced [Beveridge 93; Huttenlocher et al 93]. All of these techniques in turn require accurate camera models and accurate 3D object models, two more areas in which recent progress has been made; see [Tsai 86] for advances in camera modeling, and [Tomasi and Kanade 92] and [Oliensis 94] for advances in recovering 3D shape models from multiple images.

Of course, as we argued in Section 2.1, rigid shape matching is not the same thing as object or scene interpretation, even in 3D. It should be obvious, for example, that there is no fixed 3D shape that corresponds to the object "house". Nonetheless, the 3D matching techniques above can also be used to look for substructures within an object, such as windows or doors. In addition, other 3D techniques such as vanishing point analysis [Magee and Aggarwal 84; Collins and Weiss 90] and the perspective-angle transform [Kanatani 88] allow the 3D orientation of an object to be recovered from parallel lines on its surface or by matching a corner. Such 3D orientation information in turn allows the image to be "unwarped", removing the effects of perspective projection. In general, the burgeoning fields of projective geometric invariants [Mundy and Zisserman 92] and multiframe analysis [Ullman and Basri 91]. are giving us a greater understanding of what 3D geometric information we can and can not expect to extract from small numbers of 2D images

Finally, the improved performance of stereo algorithms at producing depth maps [Kanade and Okutami 94; Matthies 92; Schultz 94], the development of direct 3D sensors (such as LADAR, IFSAR and structured light sensors), and the introduction of depth-from-focus algorithms [Grossman 87; Krotkov 87] all provide new sources of 3D information for knowledge-directed systems. While we certainly do not claim that the 3D problem has been "solved" -- 3D shape representation is just one area in which much more work is required -- we do believe that the advancements in 3D vision over the past fifteen years will make knowledge-directed systems far more powerful. In particular, most knowledge-directed system in the past (including [Ohta 80; McKeown, et al 85; Hwang, et al 86]) were applied to aerial image domains whose near-nadir views were effectively two-dimensional[6]; we believe that restriction should no longer apply, even to complex scene interpretation systems.

3.3 Systems Integration

Integrating these new procedures into knowledge-based systems will not be easy, however. Based on our experience with the Schema System, when new vision procedures are developed, integrating them presents yet another set of problems, particularly if the new procedure was developed at another laboratory. About half the vision procedures of the time were written in C; since the VISIONS/Schema System was implemented in Lisp, this meant that half of all procedures had to be re-implemented. Even when the programming languages of the procedure and the vision system matched, the data structures rarely did. Every algorithm seemingly had its own formats for images, edges, straight lines and other commonly used geometric data structures. Applying one procedure to data created by another usually required non-trivial data conversions.

This problem has ramifications beyond knowledge-directed vision; it makes it difficult, for example, for researchers in one lab to compare their algorithms to those of another lab. As a result, the ARPA Image Understanding Environment (IUE) project [Mundy, et al. 1992] gathered a committee of noted academic researchers to create an object-oriented class hierarchy of canonical representations and file formats for every data type used in computer vision. At the time of this writing, the IUE hierarchy was fully defined, and C++ implementations of a subset of the classes were being prepared for general

[6] The VISIONS/Schema System was an exception to this rule, which is why we are so familiar with the problems of not having adequate 3D reasoning capabilities.

distribution [Dolan, et al 96]. By defining a common set of object classes for IU research, the IUE is intended to encourage and facilitate the sharing of code between laboratories.

There are three potential problems with the IUE. One is that its library of over 700? classes may omit some the representations needed to support some vision procedures[7]. Although this problem will eventually arise, the IUE class hierarchy should be large enough and general enough to cover most near-term needs. The second is that existing code designed before the IUE standard will have to be rewritten. The third is that the IUE library is so large that its implementation may prove unwieldy. However, it is clearly the most ambitious endeavor of its kind in the field of computer vision, and success would produce a revolutionary change in our ability as a field to integrate component algorithms.

Other smaller approach are also being tried, such as ISR, an object-oriented database for computer vision ([Brolio, et al 89; Draper and Kutlu 94]). Instead of defining a large class library, ISR provides a small library of frequently used classes and encourages users to define new classes as needed. The emphasis on ISR is less on the representation than on the support provided for object classes. The most important area is I/O; ISR provides an object-oriented library for I/O streams capable not only of reading and writing its own binary and ascii formats, but also those of several other computer vision systems, including Khoros [Rasure, et al 94], KBVision [Williams 90] and RCDE [Mundy, Welty, et al 92]. This minimizes the need to re-implement existing code. ISR also provides 2D and 3D graphics support, and support for spatial and associative retrieval of data tokens (class instances) from sets. Current work on ISR is focusing on integrating it with MNEME persistent object store to provide customizable mechanisms for storage/retrieval of data to/from disks, and therefore improve system performance [Kutlu,

[7] In order to keep the IUE definitions standard, researchers are discouraged from creating new classes.

et al 96]. The potential disadvantage of ISR is that as users add new classes to the system, they will diverge and the advantage of a standard representation will have been lost. To avoid this, the possibility of adopting a subset of the IUE class hierarchy as ISR's initial class library is being studied.

3.4 Tools to Support Knowledge Engineering

The VISIONS/Schema System was not the only project to encounter problems with knowledge engineering. Almost every knowledge-directed vision project encountered problems with acquiring knowledge, particularly control knowledge, and with systems integration. Artificial intelligence researchers tried to combat the knowledge acquisition problem by extracting knowledge from experts, a scenario that does not apply to computer vision. Instead, vision researchers concentrated on making knowledge easier to declare. For example, the SPAM project at CMU developed a high-level language for describing objects [McKeown, et al. 89]. Work in Japan involved both automatic programming efforts and higher-level languages for specifying image operations [Matsuyama 89]. Even today, work on improved tools for knowledge engineering continues (e.g. [Clement and Thonnat 93]).

Our experience with the Schema System led us to a different conclusion. Although projects such as the IUE and ISR are lessening the systems integration problems, we do not believe that the knowledge engineering problems can be overcome just by supplying better tools. In practice, human knowledge engineers were being relied on to create and implement complex control strategies without any scientific theory to guide them. What is needed is for a group of researchers to formalize the control problem for computer vision and to focus on its solution, just as has happened in other subareas of computer vision (e.g. edge extraction, pose determination). Based on our experience with the

Schema System, true progress in knowledge-directed vision will require a theoretically sound approach to control.

4. New Directions in Control

In a computer vision system, control is responsible for orchestrating the flow of processing within the system towards achieving a goal. The goal may be as simple as finding a single object (e.g. “find the house in this picture”), or as complex as recognizing every object in an image and determining their 3D positions. The task of the control strategy is to find the best sequence of vision procedures to satisfy the goal, where “best” may be defined in terms of accuracy, cost, or more often a combination of both.

The best strategy is a function of all the types of knowledge discussed in Section 2. Object knowledge describing the shapes, sizes, colors, textures, etc., that instances of an object class can assume is obviously central to determining which vision procedures to use. Domain knowledge is also important, both in terms of background probabilities (i.e. how many objects in this domain are green?) and in terms of sensor models and viewing conditions. Unfortunately, complete domain knowledge is rarely available. Finally, contextual knowledge about how objects relate to each other is particularly useful for broadly defined tasks such as “find all objects”, where identifying one object can provide semantic clues about what other objects to look for, as well as spatial constraints that may serve to focus attention.

In general, very little research in the field of computer vision has gone into the problem of determining the best recognition strategies. Instead, most systems are hand-crafted to achieve a particular goal, and are considered finished when some pre-set level of performance is reached, even if this performance is not optimal. Recently, however, critical new research avenues have opened up that directly address the control problem for computer vision.

4.1 Bayesian Approaches

Evidence combination was one of the topics that was intensely studied in the '70s and '80s, but it was generally not viewed as a control mechanism. With the advent of Bayesian networks [Pearl 88], however, it became possible to use Bayesian reasoning not only to combine evidence from multiple features, but to select which features to measure next. In particular, it is possible to adopt a greedy control policy which at every stage of processing selects as the next action the procedure that will have the greatest immediate impact on the system's belief in a hypothesis. This is basically the approach of SUCCESSOR [Binford et al 89], which uses Bayesian networks to control inferences over a complex, multi-level representation system based on generalized cylinders, and TEA1 [Rimey and Brown 94], which uses Bayesian networks to control an active vision system.

Bayesian networks have the advantage that they establish a clear and simple principle for selecting actions -- namely, they select the action with the strongest expected impact on a hypothesis. Unfortunately, they do not avoid the knowledge engineering problem; building a Bayesian network is a complex knowledge engineering task that involves predicting all the possible dependencies in a domain. Moreover, there are currently no robust techniques for learning Bayesian networks from experience, although this is an active area of research in artificial intelligence and machine learning.

In addition, there is no guarantee that this approach will lead to optimal or near-optimal control policies. The problem is not with Bayesian inference; as long as the topology of the network obeys the constraints of the propagation algorithm, they will accurately estimate all probabilities [Pearl 88, chapter 4]. Instead, the problem is with the greedy control policy. It is not necessarily true that selecting the action with the greatest immediate reward will lead to the best control strategy. When reasoning across multiple levels of representation (as is common; see Section 2.5), the optimal action may produce little or no immediate reward; instead, the reward in terms of increased belief may come only after several subsequent steps. The problem of selecting suboptimal actions using the greedy control heuristic has led some researchers to group short sequences of sequential actions into control primitives (e.g. TEA1).

4.2 Control as Search

Another approach is to treat the control of computer vision systems as a heuristic search problem. This introduces a variety of search techniques from the field of artificial intelligence. Jiang and Bunke [95], for example, cast computer vision as a state space search problem, in which the start state corresponds to the initial image and the goal state is the target image representation. The operators in this search space are vision procedures, each of which is described in terms of how it maps from its input to its output. Given a specific task in terms of a target representation, their system uses A* search to find the least-cost path from the initial data to the target representation.

This approach has the advantage of not requiring training data; unfortunately, it relies on having accurate predictive descriptions of vision procedures, a condition which in our experience is rarely met. To be efficient it also requires a good heuristic for the A* search algorithm, which can be viewed as another (smaller) knowledge engineering problem.

Another approach introduced by Brown and Roberts [94] is to use a genetic algorithm to search the space of possible control sequences. Brown and Roberts built a military target recognition system which identifies targets using image processing and ATR algorithms

Rather than selecting a specific sequence of algorithms, however, they specify a library of algorithms and algorithm parameters that might be used. Every possible control sequence is represented by a string specifying the algorithms and parameters to be used, and a genetic algorithm finds the best string (and therefore procedure sequence) for a given target and domain.

Both of the approaches above (state-space search and genetic algorithms) are worth pursuing in this context. However, they share a common disadvantage: they both create open-loop control policies[8]. Both systems choose a fixed sequence of procedures to be applied. As a result, if one of the procedures behaves erratically and produces an erroneous result, the system will not be able to correct for the mistake.

Chen and Mulgaonkar [92], on the other hand, infer decision trees for selecting vision procedures based in part on the results returned by previous procedures. The result is a closed-loop, rather than open-loop, control policy. Although their system was limited to reasoning at a single level of abstraction, Draper [93] introduces a system with a multi-level decision tree that is capable of reasoning across many levels of representation, including images, regions, 2D line segments, 3D planar surfaces and 3D poses (i.e. position and orientation estimates).

4.3 Markov Decision Problems

Perhaps the most promising approach is to cast computer vision as a Markov Decision Process (MDP). In an MDP, the problem domain is divided into a finite set of discrete states that form a Markov chain. The actions in an MDP are processes that cause the system to transition from one state to another according to given probability

[8] The Brown and Roberts work also uses a Bayes Net to excersize dynamic control.

distributions. Transitions also have rewards or penalties associated with them, depending on their cost and/or the state they result in. (For example, a transition that results in a goal state may have a high positive reward.) The objective in an MDP is to find a control policy that maps actions onto states so as to maximize the total expected reward (or equivalently to minimize the total expected penalty).

The classic algorithm for solving MDPs is Dynamic Programming (DP)[9], which is a successive approximation method for solving the Bellman Optimality Equation by "backing up" estimates of a value or state/value function. Unfortunately, Dynamic Programming relies on an accurate model of the system being controlled in terms of the rewards and transition probabilities, and as mentioned in Section 4.2 above, accurate models of vision procedures are rarely available. Fortunately, research in Reinforcement Learning (RL) has resulted in algorithms -- most notably Q-Learning [Watkins 89] and Temporal-Difference (TD) learning [Sutton 88] -- that use similar methods to learn control policies for unmodeled systems through on-line experimentation. This allows control policies to be learned for systems where the operators and rewards may not be well understood. (For a thoughtful discussion of RL and DP and the relationship between them, see [Barto, Bradtke et al. 96]).

Recently, Peng and Bhanu [95] applied reinforcement learning to the limited problem of parameter selection for a single vision procedure (an image segmentation algorithm). Reinforcement learning has also been used in visually-guided robotic applications. The most exciting possibility, however, is that reinforcement learning can be used to learn complex scene or image interpretation strategies.

[9] Either value-iteration or policy-iteration DP is fine. The difference is not relevant to this discussion.

In a recent version of the Schema Learning System (SLS), reinforcement learning was used to acquire a complex control policy for recognizing rooftops in aerial imagery [Draper 96]. In this experiment, SLS was provided with a library of vision procedures that included straight line extraction, pairwise line grouping algorithms and symbolic graph matching techniques. Its task was to learn a control policy for finding the image location of a rectangular roof surface in an aerial image, based on a set of training samples. In a small set of "leave one out" experiments (in which the system would be trained on nine images and tested on a tenth), the control policies learned by SLS found a correct rooftop in all ten trials (see http://vis-www.cs.umass.edu/Projects/SLS.html for more details).

The technology for applying reinforcement learning techniques to complex computer vision applications is still early in its development. Many complex questions remain, such as whether vision problems can be representation as a discrete set of states (as was done by Bhanu, et al) or whether function approximators should be used to apply reinforcement learning techniques to infinite state spaces (as was done by Draper). Nonetheless, the MDP formalization provides a rich mathematical framework for analyzing control problems in computer vision and for selecting procedures based on long-term, rather than immediate, reward.

5 Conclusion

Although most recent work in computer vision has focused on narrowly-defined subproblems, we believe that the long-terms goals of computer vision research should be more ambitious. The general goal of computer vision is to provide the sensory component - "the eyes" - for intelligent systems in a myriad of real-world applications. While the domains of interest are wide-ranging and broad, computer vision systems should be capable of providing the necessary semantic interpretations of their environment.

In many ways, the stage has been set for focusing once again on integrated systems due to the maturation of the many subfields of vision over the past twenty years. Many of the component technologies that would be used as building blocks for a general-purpose vision system are now in place. While there clearly is important work remaining in the further development of component vision technologies, it is our contention that the time is ripe to make advances in the integration of existing components into symbolic interpretation strategies.

When studying fully-integrated scene interpretation systems, one must eventually confront the many issues of knowledge representation and control, since the ultimate aim is to associate the contents of an image with semantic information stored in long-term memory. This paper reviewed some of the issues that must be dealt with, including:

- multiple levels of representation, from iconic images to extracted features, such as edges, lines, regions, textures, surfaces, volumes, object identities and spatial relationships;

- multiple visual cues and the processes to extract associated features and draw inferences from this information; this includes cues such as color, texture, occlusion, shadows, stereo, motion, shape, vanishing points, size, etc.;

- the use of context and a priori world knowledge, so that specific instances of general classes of objects can be recognized by their properties, parts of objects can be constrained by their relationship to the whole, and semantic and spatial constraints on objects within familiar stereotypical scenes can be exploited;

- the use of object- and context-specific control strategies for fusing information across multiple levels of representation and integrating evidence from different modalities.

In assessing these issues, we have argued that to fuse the many disparate visual cues available in a scene, it will ultimately be necessary to have flexible, effective, and highly-specific interpretation strategies for applying world knowledge to images. In order to take advantage of the image understanding algorithms produced over the last twenty years, these interpretation strategies can be cast as control strategies that select and parameterize sequences of vision procedures to achieve specific goals.

We contend, therefore, that one more important subproblem of computer vision needs to be focused upon and advanced before effective, general-purpose vision systems can be built: namely, the automatic construction of task-specific control strategies. Although this topic has yet received the attention we believe it deserves, early efforts to solve it are outlined here.

7. References

Andress, K. M. and A. C. Kak. (1988). “Evidence Accumulation and Flow of Control in a Hierarchical Reasoning System,” AI Magazine. Vol. 9, pp. 75-94.

Ballard, D. H., C. M. Brown and J. A. Feldman. (1978). “An Approach to Knowledge-directed image analysis” in Computer Vision Systems (A. R. Hanson and E. M. Riseman, Ed.), New York: Academic Press. .

Barto, A. G., S. J. Bradtke and S. P. Singh. (1996). "Learning to Act Using Real-Time Dynamic Programming," AI, Vol. (to appear).

Barrow, H. G. and J. M. Tenenbaum. (1978). Recovering Intrinsic Scene Characteristics From Images, AI Center, SRI International, Report# 157.

Beveridge, J. R. (1993) Local Search Algorithms for Geometric Object Recognition: Optimal Correspondence and Pose, Ph.D. thesis, University of Massachusetts, 1993.

Binford, T. O. (1982). "Survey of Model-Based Image Analysis," IJRR, Vol. 1(1), pp. 18-64.

Binford, T. O., T. S. Levitt and W. B. Mann. (1989). "Bayesian Inference in Model-Based Machine Vision" in Uncertainty in AI 3 (L. N. Kanal, T. S. Levitt and J. F. Lemmer, Ed.), New York: North Holland. .

Brolio, J., B. Draper, J. R. Beveridge and A. Hanson. (1989). "The ISR: A Database for Symbolic Processing in Computer Vision," IEEE Computer: Special Issue on Image Database Management, pp. 22-30.

Brooks, R. A. (1981). "Symbolic Reasoning Among 3-D Models and 2-D Images," AI, Vol. 17(1-3), pp. 285-348.

Brown, C. (1987). Advances in Computer Vision, Hillsdale, NJ: Erlbaum Press.

Brown, C. M. and B. Roberts. (1994). "Adaptive Configuration and Control in an ATR System," Proc. of ARPA Image Understanding Workshop, Monterey, CA, pp. 467-480.

Burns, J.B. and Riseman, E. M. (1992) “Matching Complex Images to Multiple 3D Objects using View Descriptions,” CVPR, pp. 328-334.

Canny, J. (1986) “A Computational Approach to Edge Detection,” PAMI, Vol 8(6):679-698.

Chen, C. and Mulgaonkar, P. G. (1992) “Automatic Vision Programming,” CGVIP-IU: Vol 55(2), pp. 170-183.

Clement, V. and Thonnat, M. (1993) “A Knowledge-based Approach to Integration of Image Processing Procedures,” CGVIP-IU, Vol 57, pp. 166-184.

Collins, R. T. and Weiss, R. S. (1990) “Vanishing Point Calculation as a Statistical Inference on the Unit Sphere,” ICCV, pp. 400-403.

Dolan, J., Kohl, C., Lerner, R., Mundy, J., Boult, T., and Beveridge, R. (1996) “Solving Diverse Image Understanding Problems Using the Image Understanding Environment,” IUW, pp. 1481-1504.

Draper, B., J. Brolio, R. Collins, A. Hanson and E. Riseman. (1989). “The Schema System,” IJCV, Vol. 2(3), pp. 209-250.

Draper, B. (1993). *Learning Object Recognition Strategies.* Ph.D. Thesis, Dept. of Computer Science, Univ. of Massachusetts, Amherst, MA. Also available as Technical Report 93-50.

Draper, B., G. Kutlu, A. Hanson and E. Riseman. (1994). "ISR3: Communication and Data Storage for an Unmanned Ground Vehicle," Proc. of International Conference on Pattern Recognition, Jerusalem, Israel.

Draper, B. (1996). "Learning Object Grouping Strategies for 2D and 3D Object Recognition," IUW, pp. 1447-1454.

Faugeras, O. D., Lustman, F. and Toscani, G. (1987) "Motion and Structure from Point and Line Matches," ICCV.

Grossman, P. (1987) "Depth From Focus," PRL, pp. 63-69.

Hanson, A. and E. Riseman. (1978b). "VISIONS: A computer System for Interpreting Scenes" in Computer Vision Systems (A. Hanson and E. Riseman, Ed.), New York: Academic Press. .

Haralick, R. M., Joo, H., Lee, C., Zhuang, X., Vaidya, V. G. and Kim, M. B. (1989) "Pose Estimation from Corresponding Point Data," SMC, Vol 19(6), pp. 1426-1446.

Herman, M. and T. Kanade. (1986). "Incremental Reconstruction of 3D Scenes from Multiple, Complex Images," AI, Vol. 30(3), pp. 289-341.

Hwang, V. S.-S., L. S. Davis and T. Matsuyama. (1986). "Hypothesis Integration in Image Understanding Systems," CVGIP, Vol. 36, pp. 321-371.

Hanson, A. and E. Riseman. (1987b). "The VISIONS Image Understanding System" in Advances in Computer Vision (C. Brown, Ed.), Hillsdale, NJ: Erlbaum Press. .

Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J. (1993) "Comparing Images using the Hausdorf distance," PAMI, Vol 15, pp. 850-863.

Jiang, X. Y. and H. Bunke. (1995). Vision Planner for an Intelligent Multisensory Vision System, University of Bern.

Jolion, J.-M. (1994). "Computer Vision Methodologies," CVGIP:IU, Vol. 59(1), pp. 53-71.

Kanade, T. and Okutami, M. (1994) "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," PAMI, Vol 16, pp. 920-932.

Kanatani, K. (1988) "Constraints on Length and Angle," CGVIP, Vol. 41, pp. 28-42.

Krotkov, E. (1987) "Focusing", IJCV, Vol 1, pp. 223-237.

Kumar, R. and Hanson, A. R. (1989) "Determination of Camera Position and Orientation," IUW, pp. 870-881.

Kutlu, G., Draper, B. A., and Moss, E. (1996) "Support Tools for Visual Information Management," 5th Annual Symposium on Document Analysis and Information Retrieval.

Lowe, D. G. (1985). Perceptual Organization and Object Recognition. Kluwer Academic Press, Boston.

Magee, M. J. and Aggarwal, J. K. (1984) “Determining Vanishing Points from Perspective Images,” CVGIP, Vol 26, pp. 256-267.

Marr, D. and Hildreth, E.(1980) “Theory of Edge Detection,” Proc. of the Royal Society of London, B207:187-217.

Marr, D. (1982). Vision, San Francisco: W. H. Freeman and Co.

Matsuyama, T. (1989). “Expert Systems for Image Processing: Knowledge-Based Composition of Image Analysis Processes,” CVGIP, Vol. 48, pp. 22-49.

Matthies, L. (1992) “Stereo Vision for Planetary Rovers: Stochastic Modeling for Near Real-time Implementation,” IJCV, Vol 8, pp. 71-91.

Minsky, M. (1975) “A Framework for Representing Knowledge,” in The Psychology of Computer Vision, (P. H. Winston, ed.) McGraw-Hill, N.Y.

McKeown Jr., D. M., W. A. Harvey and J. McDermott. (1985). “Rule-Based Interpretation of Aerial Imagery,” IEEE T-PAMI, Vol. 7(5), pp. 570-585.

McKeown, D. M. J., W. A. Harvey and L. E. Wixson. (1989). “Automating Knowledge Acquisition for Aerial Image Interpretation,” CVGIP, Vol. 46, pp. 37-81.

Mundy , J., T. Binford, T. Boult , A. Hanson, J. R. Beveridge, R. Haralick, V. Ramesh, C. Kohl, D. Lawton, D. Morgan, K. Price and T. Strat. (1992). “The Image Understanding Environments Program,” Proc. of DARPA Image Understanding Workshop, San Diego, CA, pp. 185-214.

Mundy, J. L., R. Welty, L. Quam, T. Strat, W. Brenner, M. Horwedel, D. Hackett and A. Hoogs. (1992). "The RADIUS common development environment," Proc. of ARPA Image Understanding Workshop, San Diego, CA, pp. 215-228.

Mundy, J.L. and Zisserman, A. (1992), eds., Geometric Invariance in Computer Vision, MIT Press, Cambridge, MA.

Nene, S., Nayar, S., and Murase, H. (1994) "Software Libary for Appearance Matching," IUW pp. 733-738.

Ohta, Y. I. (1980) "A Region-Oriented Image Analysis System by Computer", Ph.D. thesis, Kyoto University.

Oliensis, J. (1994) "A Linear Solution for Multiframe Structure from Motion", IUW, pp. 1225-1231.

Pearl, J. (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan-Kaufman Publishers, San Mateo, CA.

Peng, J. and B. Bhanu. (1995). Delayed Reinforcement Learning for Closed-Loop Object Recognition, College of Engineering, Univ. of California, Riverside.

Pomerleau, D.A., Neural Network Perception for Mobile Robot Guidance, Hingham KluwerAcademic, 1993.

Rasure, J. and Kubica, S. (1994) “The Khoros Application Development Environment,” in Experimental Environments for Computer Vision and Image Processing (H. I. Christenson and J. L. Crowley, eds.), World Scientific Press, Singapore.

Rimey, R. D. and Brown, C. M. (1994) “Control of Selective Perspection using Bayes Nets and Decision Theory,” IJCV, Vol 12, pp. 173-207.

Rosenfeld, A. (1984). “Image Analysis: Problems, Progress, and Prospects,” PR, Vol. 17(1), pp. 3-12.

Sanchez, E. and M. M. Gupta. (1984). Fuzzy Information, Knowledge Representation and Decision Analysis, Elmsford, NY: Pergamon.

Schultz, H. (1994) “Terrain Reconstruction from Oblique Views”, IUW pp. 1001-1008.

Shafer, G. and J. Pearl. (1990). Readings in Uncertain Reasoning, Morgan Kaufmann:San Mateo, CA.

Strat, T. M. (1991) “Natural Object Recognition”, Ph.D. thesis, Stanford University.

Sutton, R. S. (1988). “Learning to Predict by the Methods of Temporal Differences,” ML, Vol. 3(9), pp. 9-44.

Tarr, M.J., Black, M.J., A Computational and Evolutionary Perspective on the Role of Representation in Vision, CVGIP(60), No. 1, July 1994, pp. 65-73.

Tomasi, C. and Kanade, T. (1992) “The Factorization Method for the Recovery of Shape and Motion from Image Streams,” IUW pp. 459-472.

Tsai, R. Y. (1986) “An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision,” CVPR, pp. 364-374.

Ullman, S. and Basri, R. (1991) “Recognition by Linear Combination of Models,” PAMI, Vol 13(10), pp. 992-1006.

Walker, E. L., Herman, M., and Kanade, T. “A Framework for Representing and Reasoning about Three-Dimensional Objects for Vision,” AI Magazine, 9(2):47-58.

Wallace, A. M. (1988). “A Comparison of Approaches to High-Level Image Interpretation,” Pattern Recognition, Vol. 21(3), pp. 241-259.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards.* Ph.D. Thesis, Cambridge Univ., Cambridge, England,

Wesley, L. (1986) “Evidential Knowledge-Based Computer Vision,” Optical Engineering, Vol 25(3), pp. 363-379.

Williams, T. D. (1990). “Image understanding tools,” Proc. of International Conference on Pattern Recognition , Vol. D, pp. 606-610.

Zimmermann, H. J. (1985). Fuzzy Set Theory and its Applications, Boston, MA: Kluwer.