Employing Contextual Information in Computer Vision

Thomas M. Strat
Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

Abstract

Contextual information is often essential for visual recognition, but the design of image-understanding systems that effectively use context has remained elusive. We describe some of our experiences in attempting to employ contextual information in computer vision systems. By making explicit the built-in assumptions inherent in all computer vision algorithms, an architecture can be designed in which context can influence the recognition process. This paper describes such an architecture for context-based vision (CBV).

1 Introduction

It is generally accepted that the surroundings of an object may have a profound influence on, and in some cases, may be necessary for, visual recognition of the object. What is not so well established is how to design computer vision systems that can exploit such contextual information.

When a human observes a scene, or even studies a photograph, he normally has at his disposal a wealth of information that is not captured by the image alone. For example, if Bob shows Alice some photographs he took, her knowledge that Bob recently vacationed in Hawaii may help her to recognize that the photos were taken there. Any knowledge that Alice has about Hawaii may be useful for recognizing the content of the scene (e.g, that the amorphous landform is actually Diamond Head, and that the vegetation is palmetto bushes and not agave cacti).

An observer can also infer information about the scene that is then useful for interpreting other parts of the image. For example, given an outdoor scene, usually one can readily determine where the sky is, which direction is vertical, what the weather conditions are, and whether any man-made objects are visible. This information forms part of the context that is available for interpreting the remainder of the scene.

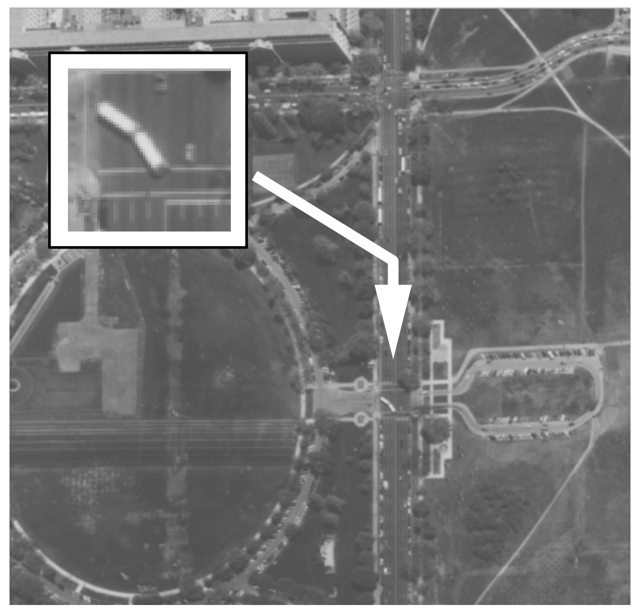

Figure 1: An image in which the use of context is critical to the recognition of some objects.

An image such as shown in Figure 1 illustrates the power of contextual information. The inset, a magnified portion of the larger image, displays an object that is difficult to recognize. When the same object is viewed in the context of the intersection of city streets (as in the large image), it is readily recognized as an articulated bus.

[1]The work reported here was sponsored by DARPA and monitored by the US Army Topographic Engineering Center under Contract DACA76-92-C-0034.

In this paper, we describe some of our experiences in attempting to employ contextual information in computer vision systems. By making explicit the built-in assumptions inherent in all computer vision algorithms, an architecture can be designed in which context can influence the recognition process. This paper describes such an architecture for context-based vision (CBV).

The first half of the paper summarizes the types of contextual information that are available in image-understanding systems and describes some roles that context can play in the interpretation process. The second half reviews a previously constructed context-based architecture, CONDOR; describes some extensions that are necessary to extend its applicability to semiautomated image understanding (IU); and presents some empirical results of its use in extracting cartographic features.

2 Context-Based Vision

We use the term *contextual information*, or *context* for short, in the broadest sense — to denote any and all information that may influence the way a scene is perceived. Thus, the camera geometry, the image type, the availability of related images, the urgency of observation, and the purpose of image analysis, are all part of the context. A computer vision system, like a human, should be able to use all types of context.

Many authors have used contextual information either implicitly or explicitly in their IU systems, but few have made the representation and use of context a central design feature [4, 5, 7, 13, 21].

The effective use of contextual information can be addressed by considering the design of an overall system architecture, rather than by focusing on individual algorithms. In our view, this can be accomplished by structuring a computer vision system as a composite of many individual algorithms. The contextual information, including the perceptual task and the available imagery, can be used to choose the algorithms most appropriate for each subtask, and can form the basis for evaluating their results. The algorithms can perform independently, but are able to interact through the context that all are controlled by and all contribute to.

The concepts described in this paper are illustrated by examples from two architectures we have designed:

- CONDOR [17, 18, 19] is a system that analyzes ground-level outdoor imagery of natural environments in the context of a mobile robot application. CONDOR contains an elaborate mechanism for recognizing and labeling natural objects automatically. Because natural objects, unlike man-made objects, are difficult to recognize without consideration of context, analysis of these scenes demands an architecture that makes strong use of contextual information.
- The second architecture is being developed as part of a system for site-model construction using overhead imagery in the RADIUS Project [8]. Unlike CONDOR, this system is designed to be semiautomated — a fact that has implications for both the way in which context can be employed, and for the availability of contextual information. Being a semiautomated design, it relies upon a human operator to replace some of the machinery incorporated in CONDOR and exploits additional contextual constraints supplied by the operator.

3 The Need for Context

The technical problems in using context involve the identification of appropriate representations for the relevant knowledge and the design of an architecture that can effectively invoke this knowledge. A context-based architecture for image understanding must have (among other things) a means for enforcing the assumptions of IU algorithms and a means for accessing relevant information.

3.1 Enforcing Assumptions

Every image-understanding algorithm, by necessity, contains numerous built-in assumptions that limit its range of applicability. For example, some edge-finders work only on binary images, some stereo algorithms cannot handle occlusions, and some road-finders are confounded by strong shadows.

If the results of these algorithms are to be relied upon, the algorithms must not be employed in situations for which their designers did not intend them to be used. It is the context of invocation that dictates the suitability of an algorithm for a particular task. By explicitly encoding the assumptions and inherent limitations of IU algorithms, one has the potential to control the algorithms by reasoning about the context. Representing assumptions explicitly and matching them to the particular circumstances is one of the keys to using contextual information in a computer vision system.

3.2 Accessing Nonlocal Information

Most IU algorithms also require the use of nonlocal information — data outside the immediate sphere of computation — to assist the interpretation or to control the processing flow. Examples include pixel data that are outside some local processing window, additional images of the same scene, prior facts or expectations that are stored in a map or database, and generic knowledge about the appearance, function, or purpose of objects in a scene. Such information is used by many IU algorithms to compute parameters, to guide search, to cue recognition processes, or to reason about the consistency of an interpretation.

IU algorithms must have access to nonlocal information to aid interpretation. Providing direct access to relevant nonlocal information is another key to using contextual information in a computer vision system.

4 Types of Context

Before describing how contextual information can be represented and used, it is useful to take inventory of the kinds of context that could be considered.

Figure 2 depicts a schematic view of an IU algorithm as a black box. Its explicit inputs are a set of images and some parameters, but it is invoked in the context of an assigned task, a database of facts about the world, and a knowledge base from which additional information about the world can be deduced. Some of its outputs are symbolic descriptions that can also be used to augment the database or knowledge base, or to assign additional tasks for realizing behaviors.

We have found it convenient to divide the range of contextual information into three categories. Additional semantic knowledge may involve contextual information from all three categories.

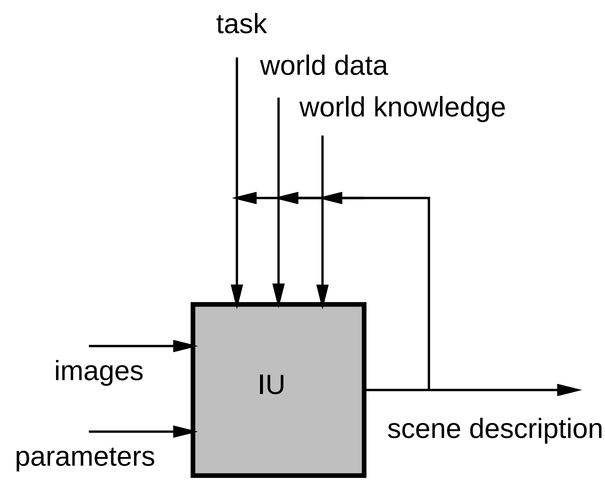

Figure 2: A schematic diagram of an IU algorithm embedded in a vision system.

Physical context — information about the visual world that is independent of any particular set of image acquisition conditions. Physical context encompasses a range of specificity from the very precise "There is a tree at (342, 124)" to the more generic "This area contains a mixed, deciduous forest." Physical context may also include information about the appearance of scene features in previously interpreted imagery and dynamic information, such as weather conditions and seasonal variations.

Photogrammetric context — information surrounding the acquisition of the image under study. This includes both internal camera parameters (*e.g.*, focal length, principal point, field of view, color of filter) as well as external parameters (*e.g.*, camera location and orientation). We also include the date and time of image acquistion as well as the images themselves.

Computational context — information about the internal state of processing. The computational context can be used to control the processing sequence based on partial recognition results. Different strategies can be used when first initiating the analysis of an image versus filling in the details of a largely completed analysis. The assigned task, the level of automation required, and the available hardware processes are all construed as part of the computational context.

It is worth noting that context may be either established or hypothetical. Tentative conclusions such as "The sky is not visible in this image," and hypothesized facts about the world such as "Assuming that no buildings with peaked roofs are at this site" can be treated as ordinary context to generate hypothetical conclusions.

Just what constitutes contextual information is highly dependent upon the domain of application and the goals of the image-understanding system. CONDOR and RADIUS both involve the delineation and recognition of features of the outdoor world from multiple images. Tables 1–3 detail the types of context used or usable in these applications.

The information in the tables was compiled by examining about one hundred IU algorithms embedded in CONDOR. That list was then augmented by considering additional algorithms that appear to be relevant to the RADIUS site-model construction application. The algorithms considered range from edge-finders [1] to image-segmentation [12], to stereo compilation [2], to snakes [10], to complete object recognition systems [3, 20]. The associated parameters and implicit assumptions for each algorithm were tabulated.

Contextual information may come from a variety of sources, depending on the nature of the application. Some representative sources of contextual information are

- Database – Information for use by a vision system may have been previously compiled and stored. Geometric object models, map data, and iconic texture maps are examples.

- Image header – Information about the image acquisition is often stored with the image. Camera models, image size and type, and time and date of acquisition are examples.

- Derived – Results of earlier IU computation are a valuable source of additonal information about a scene.

- User – In an interactive or semiautomated scenario, the human operator is also a source of information that can provide context to IU algorithms. This information could range from a general characterization of the image (*e.g.*, urban environment) to a precise, manual extraction of individual features.

Table 1: Physical Context

Geometry	Geometric models of roads, trails, fences, trees, rocks, buildings, railroads, towers, fields, etc. 3D Outline Location Orientation
Photometry/ Radiometry	Albedo Material type Reflectance Surface properties Previous image snippets
Illumination	Sun (azimuth, elevation angles) Haze Cloud cover Shadow contrast
Weather	Temperature Current Precipitation Recent Precipitation Wind speed and direction Season
Geography	Site Terrain type (tundra, desert, ocean, ...) Land use (urban, rural, agricultural, ...) Topography (*e.g.*, Digital Elevation Model) Environmental events (fire, flood, earthquake, war, ...)
Other	Semantic properties (name, use, history, ...)

5 Uses of Context

When an IU algorithm is viewed as a black box as in Figure 3, it is apparent that there are only two opportunities to use contextual information to influence its behavior. At the input end, context can be used to select the best match of image data with IU algorithms and their parameters. At the output end, context can be used to analyze and filter the results.

Choosing algorithms and their parameters: Given an image and a task to be performed, it is necessary to determine the most appropriate algorithm or set of algorithms for accomplishing the task. When the assumptions and limitations of each algorithm have been coded explicitly, it is possible to match their requirements with the context of the present situation, and choose the ones that have (at least) the potential to achieve the desired result. Similarly, a mechanism can be constructed to compute the parameters associated with those algo-

Table 2: Photogrammetric Context

Date and time	
Look angle	Azimuth, elevation, roll
Footprint	Portion of ground observed
Modality	Infrared, color, radar, ...
Multiplicity	Monocular, binocular stereo, multiple, ...
Image size	Pixel dimensions
Image element type	Binary, scalar, vector, complex, ...
Resolution	Ground sample distance (GSD)
Camera model	Focal length, principal point, nonperspective, ...

Table 3: Computational Context

Task	Interpret everything, find tanks, model all buildings, ...
Interactivity	Fully automatic, manual, semiautomatic, batch, continuous interaction, ...)
Urgency	Acceptable processing time
Hardware	Uniprocessor, special-purpose hardware, multiprocessor, ...
Processing state	Just starting, already looked, detailed search, ...

rithms from the available context, although it may be difficult to identify the appropriate computations in advance.

Choosing image data: In some applications, including the CONDOR and RADIUS scenarios, a multitude of imagery is available for analysis. Choosing the subset of images to use can be as critical as the selection of appropriate algorithms. When an algorithm is being considered for invocation, the explicitly coded assumptions can be used to select the images that are best suited to the extraction task being given to that algorithm.

Evaluating results: When IU algorithms have completed their processing, the system has produced a set of results that are best considered as hypotheses. Analysis of the results with the benefit of relevant contextual information can lead to improved interpretations of the imagery. This analysis can take place in several ways — by ranking the hypotheses, by comparing them, by checking their consistency with other hypotheses or with the established context, and so on. In each case, if the analysis software is encoded as a collection of algorithms with explicitly encoded assumptions, one can use the context to choose the algorithms and

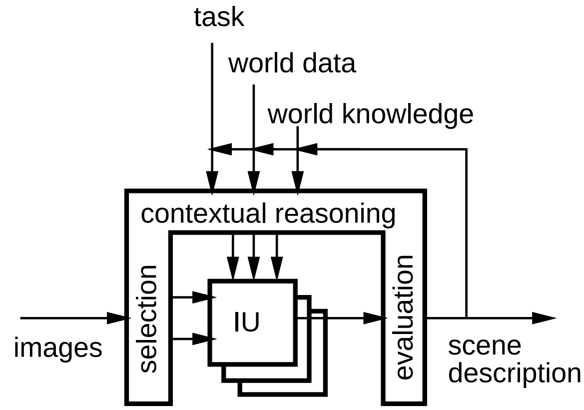

Figure 3: A schematic diagram of a context-based vision system.

control their invocation. Not only does this approach reduce unnecessary computation, but it also simplifies software construction because each algorithm need work only in some narrowly defined context.

6 An Architecture for Context-Based Vision

In the context-based vision paradigm, the invocation of all algorithms is governed by context. Rather than having the control structure and control decisions to be made hard-wired, the process is driven by context.

CONDOR was designed as the perceptual architecture for a hypothetical outdoor robot. Given an image and a possibly extensive database describing the robot's environment, the system is to analyze the image and to augment the world model. CONDOR's recognition vocabulary consists mainly of natural objects such as trees, bushes, trail, and rocks. Because of the difficulty of recognizing such objects individually, CONDOR accepts an interpretation only if it is consistent with its world model. CONDOR recognizes entire contexts, rather than individual objects [17, 18, 19].

6.1 Context Sets

We associate a data structure called a *context set* with each IU algorithm. The context set identifies those conditions that must be true for that algorithm to be applicable. Efficient and effective vi-

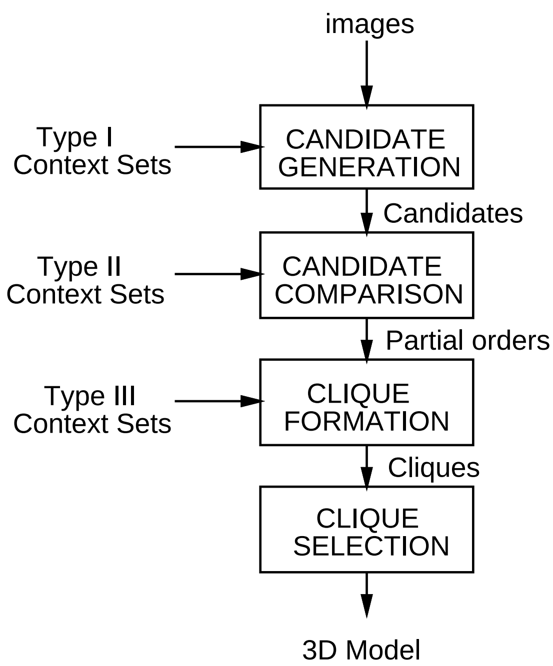

Figure 4: Sequence of Computation in CONDOR.

sual recognition can be achieved only by invoking the IU algorithms in those contexts in which they are likely to succeed.

Formally, a context set is a collection of context elements that are sufficient for inferring some relation or applying some algorithm. A *context element* is a predicate involving any number of terms that refer to the physical, photogrammetric, or computational context of image analysis.

Each algorithm has an associated context set, and is invoked only if its context set is satisfied. A context set is considered to be satisfied only if all its context elements are satisfied.

As an example, consider a simple operator that extracts blue regions to find areas that could be labeled "sky." A context set for this operator might be

{ image-is-color, camera-is-horizontal, sky-is-clear, time-is-daytime }

The blue-sky algorithm would be unreliable if it were employed in anything but this context.

6.2 Approach

The CONDOR architecture employs three types of algorithms controlled by context sets, as illustrated in Figure 4:

- Type I context sets control IU algorithms that produce candidate (hypothetical) labeled regions.
- Type II context sets control algorithms that compare two candidates and determine if one should be preferred over the other. This step is mainly necessary to limit the combinatorics of finding mutually consistent candidates.
- Type III context sets control algorithms that check if a candidate is consistent with an emerging world model.

For each class in the active recognition vocabulary, all Type I context sets are evaluated. The operators associated with those that are satisfied are executed, producing candidates for each class. Type II context sets that are satisfied are then used to evaluate each candidate for a class, and if all such evaluators prefer one candidate over another, a preference ordering is established between them. These preference relations are assembled to form partial orders over the candidates, one partial order for each class. Next, a search for mutually coherent sets of candidates is conducted by incrementally building cliques of consistent candidates, beginning with empty cliques. A candidate is nominated for inclusion into a clique by choosing one of the candidates at the top of one of the partial orders. Algorithms associated with Type III context sets that have been satisfied are used to test the consistency of a nominee with candidates already in the clique. A consistent nominee is added to the clique; an inconsistent one is removed from further consideration with that clique. Further candidates are added to the clique until none remain. Additional cliques are generated in a similar fashion as computational resources permit. Ultimately, one clique is selected as the best interpretation of the image on the basis of the portion of the image that is explained and the reliability of the operators that contributed to the clique.

The interaction among context sets is significant. The addition of a candidate to a clique may provide context that could trigger a previously unsatisfied context set to generate new candidates or establish new preference orderings. For example, once one bush has been recognized, it is a good idea to look specifically for similar bushes in the image. This tactic is implemented by a candidategeneration context set that includes a context element that is satisfied only when a bush is added to a clique.

6.3 Representation of Context

We have outlined a paradigm in which the requirements of algorithms are matched against the context of a given situation. To employ this paradigm, it is necessary to have representations for the various categories of contextual information that are to be employed.

The CONDOR system employs the Core Knowledge System (CKS), an object-oriented knowledge/database that was specifically designed to serve as the central information manager for a perceptual system [15]. The CKS provides the ability to store contextual information, and to retrieve it through a vocabulary of spatial and semantic queries. It has the further ability to accommodate conflicting data from multiple sources without corrupting the inference channels. CONDOR uses CKS to store a persistent model of the world, and then uses that model as context for image understanding. Image-understanding results are stored in the CKS and hence are available as context for subsequent processing.

The SRI Cartographic Modeling Environment (CME) provides the primitive representations for modeling the physical objects and their attributes [9]. CME is also used for geometric operations, including coordinate transformation, and for display of imagery and synthetically generated scenes.

6.4 Results

Figure 5(a) depicts an image that typifies those analyzed by CONDOR. After several thousand IU algorithm invocations and construction of 20 cliques, CONDOR's best clique correctly identified six of the trees visible in the image. A perspective view of the grass and trees in the 3D model produced by CONDOR is shown in Figure 5(b).

CONDOR was able to achieve similar results from processing more than 100 images of natural scenes taken within a limited 2-square-mile area. When tasked to analyze images from other natural areas, CONDOR's performance degrades because its contextual knowledge is not totally relevant. This simultaneously illustrates the power of using context, as well as the need to encode all contextual constraints that are likely to arise.

7 RADIUS – Site Model Construction

We now turn our attention to the RADIUS project, which is concerned with constructing site models of cultural objects from overhead imagery. Although the specific algorithms to be employed in RADIUS are likely to differ greatly from those in CONDOR, their demands for contextual information are very similar.

The biggest difference between CONDOR and RADIUS is the fact that RADIUS is being designed as a semiautomated system. Accordingly, our design chooses to leave the evaluation of IU results to the human operator. As a result, the Types II and III context sets employed in CONDOR are not necessary. Instead, we concentrate on the construction of Type I context sets for controlling the invocation of IU algorithms. This is particularly appropriate for RADIUS given the wide variety of features to be extracted and the large number of IU laboratories expected to contribute algorithms.

The examples presented here are drawn from an architecture that is being designed to support site model construction for the RADIUS application. The architecture incorporates a large number of generic cartographic feature extraction algorithms; it uses contextual information to identify those most likely to succeed at a given task and to set their associated parameters.

7.1 Model-Based Optimization

While the architecture we have designed is capable of enforcing the contextual constraints of almost any IU algorithm, our initial experiences have focused primarily on employing algorithms from a paradigm known as Model-Based Optimization (MBO).

Specializations of MBO have been referred to by various other terms, including dynamic programming [6], regularization [14], deformable surfaces [22], and snakes [10]. The approach underlying MBO is to express the solution to a feature-extraction problem as a mathematical function of

Figure 5: Example of Processing Results by CONDOR.

some variables, and then to extract the feature from imagery by adjusting the values of the variables to minimize the function. Typically the objective function includes terms that bias the feature's geometry as well as its match with image data. As we have posed it, MBO operators require four parameters: topological primitive, objective function to be minimized, source of initial conditions, and the optimization procedure to be employed. The Context-Based Vision architecture must set these parameters on the basis of known contextual information or (in some cases) human input.

7.2 Context Sets

In CONDOR, Type I context sets are used to specify the conditions that must be met for a given algorithm to be applicable. The context set can also specify the conditions that must be met for a given parameter setting to be useful. For example,

MBO(closed-curve, rectangular-corners,
manual-entry, gradient-descent):

specifies the parameters for an MBO algorithm that could be used to extract roof boundaries under some circumstances. The following context set encodes conditions that are required for the extraction of roofs using that algorithm:

{ image-is-bw, image-resolution≤ 3.0,
interactivity-is-semiautomated }

This context set gives the requirements that must exist for the above MBO algorithm to be applicable and it specifies the suitable parameter values. In the example above for detecting roofs, these parameters have been specified as a closed-curve topology, an objective function preferring rectangular corners, initial boundary provided by manual entry, and the use of a gradient-descent optimization procedure.

In practice, a large number of context sets governing the application of MBO algorithms as well as other algorithms could be constructed and used to implement a cartographic feature-extraction system suitable for site-model construction. It is clear that such a collection could be unwieldy and difficult to maintain. A more structured representation of the context set concept is needed.

7.3 Context Tables

One alternative representation for context sets is the *context table* — a data structure that tabulates the context elements in a more structured fashion. An IU algorithm is associated with each row in the table; each column represents one context element.

The context table is equivalent to a collection of context sets. Conceptually, it provides a more coherent view of the contextual requirements of related algorithms. Applicable algorithms are selected by finding rows for which all conditions are

Table 4: A Context Table

	feature	interactivity	images	resolution	geography	algorithm
1	roof	semiautomated	single BW	≤ 3 meters	—	MBO(topology=closed-curve, obj-fn=rectangular-corners, init=manual-entry, opt=gradient-descent)
2	roof	manual	single	≤ 10 meters	—	CME(primitive=closed-curve)
3	road	semiautomated	single BW	≤ 1 meter	hilly	MBO(topology=ribbon-curve, obj-fn=(smoothness(0.5),continuous, parallel), init=manual-entry, opt=gradient-descent)
4	road	semiautomated	single BW	≤ 10 meters	hilly	MBO(topology=open-curve, obj-fn=(smoothness(0.5),continuous), init=manual-entry, opt=gradient-descent)
5	road	semiautomated	single BW	≤ 1 meter	flat $\vee$ urban	MBO(topology=ribbon-curve, obj-fn=(smoothness(0.8),continuous, parallel), init=manual-entry, opt=gradient-descent)
6	road	semiautomated	single BW	≤ 10 meters	flat $\vee$ urban	MBO(topology=open-curve, obj-fn=(smoothness(0.8),continuous), init=manual-entry, opt=gradient-descent)
7	road	manual	single	≤ 10 meters	—	CME(primitive=open-curve)
8	road	manual	single	≤ 1 meter	—	CME(primitive=ribbon-curve)
9	road	semiautomated	single	≤ 2 meters	—	ROAD-TRACKER (control=bidirectional-search, init=manual-entry)

met. Table 4 contains an excerpt of a context table for use in cartographic feature extraction which illustrates the representation.

One drawback to the table representation is its potentially large size. Each algorithm may require many rows to capture the contextual constraints of its various parameter combinations. Its chief value is its organization of contextual information for knowledge-base construction.

7.4 Context Rules

A third alternative for representing context sets is to encode them as rules whose antecedent is the context set, and whose consequent is the applicable algorithm.

For example,

{ image-is-bw, image-resolution≤ 3.0,
interactivity-is-semiautomated } $\Longrightarrow$
MBO(closed-curve, rectangular-corners,
manual-entry, gradient-descent):

One advantage of encoding the rules as a logic program is that using the logic program interpreter eliminates the need to devise special machinery to test satisfaction of context sets. The context table of the previous section (Table 4) can be recoded as the roughly equivalent Prolog program given in the Appendix.

A further representational efficiency is possible by collapsing rules with common context elements. For example. the only difference between rules governing Algorithms 3 and 4 and rules governing Algorithms 5 and 6 is the geography term and the value of the smoothness parameter. This dependence could be generalized by additional rules that relate smoothness to geography.

Whatever representation is chosen, it is clear that context sets can be employed in either direction. In the forward direction, the context sets are used to find applicable algorithms. In the opposite direction, the sets can be used for several purposes, including the selection of images on which to invoke a given algorithm. For example, Table 4 shows that the use of an MBO algorithm for finding a roof (Row 1) requires the existence of a monochrome image with 3-meter resolution or better.

7.5 Results

Although the architecture we have described for the RADIUS application is not yet fully functional, we can illustrate its application using the example Table 4.

Figure 6 compares the results of applying an MBO algorithm both within and outside its inherent contextual constraints. Figure 6(a) shows an overhead view of a portion of the Mall in Washington, DC — a flat park area in an urban setting. Figure 6(b) shows an overhead image of a hilly area in the foothills of the Rocky Mountains in Colorado. Both images are shown at approximately the same scale.

The context table in Table 4 can be used to select an algorithm suitable for extracting roads in a semiautomated setting. In the context of the analysis of the Washington DC image, both Algorithm 5 and Algorithm 9 are applicable, but we ignore Algorithm 9 in this example. Algorithm 5 calls for manual entry of the initial curve, which is shown in Figure 6(a). Optimization of this curve using the specified objective function and optimization procedure results in the model depicted in Figure 6(c) — a reasonably accurate extraction of the road.

This algorithm is not applicable to the Rocky Mountain image, because of the different geographical context. If it were applied anyway, optimization of the initial curve shown in Figure 6(b) would result in the curve shown in Figure 6(d) — an extraction that does not follow the road boundaries well.

The context table shows that Algorithm 3 (with its lower smoothness parameter) is applicable for the Rocky Mountain image. Applying it to the same initial curve gives the result depicted in Figure 6(f), a significant improvement over that obtained by Algorithm 5.

Had Algorithm 3 been appplied to the Washington DC image (where its context is violated), the result shown in Figure 6(e) would have been obtained – a noticeably poorer delineation of the road than that obtained with a higher smoothness parameter. It is not surprising that the choice of parameters can have a critical effect on the output of an IU algorithm. More important, this example illustrates that contextual information can be successfully used to choose parameter settings.

7.6 Knowledge-Base Construction

The context sets (or context table or context rules) constitute the knowledge base employed by the system. It is clear that the performance of the system will be limited by the accuracy and completeness of the knowledge base. The context sets employed in CONDORand the context rules being constructed for the RADIUS application are hand-crafted based on *ad hoc* experimentation with available imagery. It is clear that a more automated, or at least a better grounded procedure for constructing the context rules is desirable, both for accommodating a potentially large knowledge base and for extending the domain of competence beyond that originally conceived.

There are several approaches by which the system could learn the most effective context rules. Perhaps the most enticing one for interactive interpretation is one in which the system learns through experience. Whenever a situation arises for which there is no applicable algorithm, or for which all the applicable algorithms give unacceptable results, the human operator has no choice but to edit the result or model the feature by hand, and then continue the site-model construction. Such a manual extraction can serve as the "correct" answer in a supervised learning process. By capturing the context that failed initially, the learning procedure can theoretically compare the results of many algorithms with the "correct" one — whenever there is a sufficiently accurate match, a new context rule can be added. One can also imagine finding a better set of parameters by posing the problem as one for MBO: the algorithm's parameters can be varied systematically until the best match with "correct" answer is obtained. If the match is sufficiently close, a new context rule with the corresponding parameter settings can be installed.

Automating the construction of the context rules is both important and difficult. There are many promising approaches, but none have yet been seriously tried.

8 Summary

We have described some of our experience in applying the CONDOR architecture to the site-model construction task of RADIUS. The semiautomated nature of RADIUS obviates the need for some of the machinery employed in the fully automated design of CONDOR. The availability of a human operator permits access to some kinds of context that were not available to CONDOR, such as the level of interactivity desired, and manual sketches of individual features. The existence of a human to review and edit the IU results offers the opportunity to use a supervised learning scheme to improve the quality of the knowledge base or to extend its range of competence.

The large number of features and wide range of imaging conditions that must be considered for site-model construction in RADIUS stress the context set representation employed in CONDOR. While context sets were adequate for the knowledge base of CONDOR, it has been necessary to consider more effective representations that will extend to the requirements of site-model construction. Two new constructs — context tables and context rules — offer a more systematized organization for the context knowledge base that should facilitate its construction. These representations offer additional economies in both storage and computation that may be vital to implemetation of large systems. The symmetry of context tables and rules encourages their use in either direction: to select algorithms and set their parameters, or to describe the conditions that must be satisfied for a given algorithm to be applicable. This final capability raises the possibility of using context rules to choose the most appropriate images for interpretation.

Acknowledgments

I am indebted to Marty Fischler for the numerous discussions that motivated and shaped much of this work. Thanks also to Pascal Fua for the use of his snake algorithms and to Lynn Quam for supplying the Cartographic Modeling Environment which facilitated the implementation and experimentation enormously.

Appendix

The following Prolog program[2] is roughly equivalent to the context table depicted in Table 4.

[2] More compact programs are possible.

```
% alg(Name, Parameters) :-

% alg specifies the applicable functions and their
%    appropriate parameter settings for use in a
%    prescribed context
%  Name is a symbol denoting the function to be invoked
%  Parameters is a sequence of parameters whose format
%    depends on the function

alg(mbo, [closed-curve, obj-fn(rectangular-edges),
    manual-entry, gradient-descent]) :-
  object-type(roof),
  site(Site),
  interactivity(semiautomated),
  image-site(Image,Site),
  modality(Image, bw),
  image-resolution(Image, GSD),
  GSD =< 3.0 .

alg(cme, [closed-curve] ) :-
  object-type(roof),
  site(Site),
  interactivity(manual),
  image-site(Image,Site),
  image-resolution(Image, GSD),
  GSD =< 10.0 .

alg(mbo, [ribbon-curve,
    obj-fn(smoothness(0.5), continuous, parallel),
    manual-entry, gradient-descent]) :-
  object-type(road),
  site(Site),
  interactivity(semiautomated),
  image-site(Image,Site),
  modality(Image, bw),
  image-resolution(Image, GSD),
  GSD =< 1.0,
  site-geography(Site, hilly) .

alg(mbo, [open-curve,
    obj-fn(smoothness(0.5), continuous),
    manual-entry, gradient-descent]) :-
  object-type(road),
  site(Site),
  interactivity(semiautomated),
  image-site(Image,Site),
  modality(Image, bw),
  image-resolution(Image, GSD),
  GSD =< 10.0,
  site-geography(Site, hilly) .


alg(mbo, [ribbon-curve,
    obj-fn(smoothness(0.8), continuous, parallel),
    manual-entry, gradient-descent]) :-
  object-type(road),
  site(Site),
  interactivity(semiautomated),
  image-site(Image,Site),
  modality(Image, bw),
  image-resolution(Image, GSD),
  GSD =< 1.0,
  site-geography(Site, flat) .

alg(mbo, [open-curve,
    obj-fn(smoothness(0.8), continuous),
    manual-entry, gradient-descent]) :-
  object-type(road),
  site(Site),
  interactivity(semiautomated),
  image-site(Image,Site),
```

```
  modality(Image, bw),
  image-resolution(Image, GSD),
  GSD =< 10.0,
  site-geography(Site, flat) .

alg(cme, [open-curve] ) :-
  object-type(road),
  site(Site),
  interactivity(manual),
  image-site(Image,Site),
  image-resolution(Image, GSD),
  GSD =< 10.0 .

alg(cme, [ribbon-curve] ) :-
  object-type(road),
  site(Site),
  interactivity(manual),
  image-site(Image,Site),
  image-resolution(Image, GSD),
  GSD =< 1.0 .

alg(road-tracker,
    [bidirectional-search, manual-entry] ) :-
  object-type(road),
  site(Site),
  interactivity(semiautomated),
  image-site(Image,Site),
  image-resolution(Image, GSD),
  GSD =< 2.0 .
```

References

[1] Ballard, Dana H., and Christopher M. Brown, *Computer Vision*, Chapter 3, Prentice-Hall, Inc., 1982.

[2] Barnard, Stephen T., and Martin A. Fischler, "Computational Stereo," *Computing Surveys* **14**(4), pp. 553–572 (December 1982).

[3] Binford, Thomas O., "Survey of Model-Based Image Analysis Systems," *Int. J. Robotics Research*, Vol. 1, pp. 18–64 (1982).

[4] Clément, Véronique, Gérard Giraudon, Stéphane Houzelle, and Fadi Sandakly, "Interpretation of Remotely Sensed Images in a Context of Multisensor Fusion using a Multi-Specialist Architecture," Institut National de Recherche en Informatique et en Automatique (INRIA), Research Report, Sophia Antipolis, France (October 1992).

[5] Draper, Bruce A., Robert T. Collins, John Brolio, Allen R. Hanson, and Edward M. Riseman, "The Schema System," *Internat. Jour. Computer Vision,* **2**(3), pp. 209–250 (January 1989).

[6] Fischler, M.A., and R.A. Elschlager, "The Representation and Matching of Pictorial Structures," *IEEE Trans. on Comp.* **22**(1), pp. 67–92 (January 1973).

[7] Garvey, Thomas D., "Perceptual Strategies for Purposive Vision," Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, Stanford, CA (December 1975).

[8] Gerson, Donald J., "RADIUS: The Government Viewpoint," Proceedings, DARPA Image Understanding Workshop, San Diego, CA, pp. 173–176 (January 1992).

[9] Hanson, Andrew J., and Lynn Quam, "Overview of the SRI Cartographic Modeling Environment," *Proceedings: DARPA Image Understanding Workshop*, Cambridge, MA, pp. 576–582 (April 1988).

[10] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models," *Internat. Jour. Computer Vision,* **1**(4), pp. 321-331 (1987).

[11] Lawton, D.T., T.S. Levitt, C. McConnell, and J. Glicksman, "Terrain Models for an Autonomous Land Vehicle," in *Proc. 1986 IEEE International Conference on Robotics and Automation,* San Francisco, CA, pp. 2043 – 2051 (April 1986).

[12] Y. G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning," *Internat. Jour. Computer Vision* **3**(1), pp. 73-102 (May 1989).

[13] McKeown, David M., Jr., W.A. Harvey, Jr., and J. McDermott, "Rule-Based Interpretation of Aerial Imagery," *IEEE Trans. Pattern Analysis and Machine Intelligence* **7**(5), pp. 570–585 (September 1985).

[14] Poggio, Tomaso, Vincent Torre, and Christof Koch, "Computational Vision and Regularization Theory," *Nature* **317**(26), pp. 314–319 (September 26, 1985).

[15] Strat, Thomas M., and Grahame B. Smith, "The Core Knowledge System," Technical Note 426, Artificial Intelligence Center, SRI International, Menlo Park, CA (October 1987).

[16] Strat, Thomas M., and Grahame B. Smith, "A Knowledge-Based Information Manager for Autonomous Vehicles," Chapter 1 in Su-shing Chen (Ed.), *Image Understanding in Unstructured Environments*, World Scientific Publishing Co., Singapore, pp. 1–39 (1988).

[17] Strat, Thomas M. and Martin A. Fischler, "Natural Object Recognition: A Theoretical Framework and Its Implementation," in *Proceedings, IJCAI-91,* Sydney, Australia, pp 1264–1270 (August 1991).

[18] Strat, Thomas M., and Martin A Fischler, "Context-Based Vision: Recognizing Objects Using Both 2D and 3D Imagery," *IEEE Trans. Pattern Analysis and Machine Intelligence* **13**(10), pp. 1050–1065 (October 1991).

[19] Strat, Thomas M., *Natural Object Recognition,* Springer-Verlag, New York, 1992.

[20] P. Suetens, P. Fua, and A.J. Hanson, "Computational Strategies for Object Recognition," *ACM Computing Surveys* **24**(1) (March 1992).

[21] Tenenbaum, Jay M., Harry G. Barrow, Robert C. Bolles, Martin A. Fischler, and Helen C. Wolf, "Map-Guided Interpretation of Remotely-Sensed Imagery," *Pattern Recognition and Image Processing,* pp. 610–617 (February 1979).

[22] Terzopoulos, D., A. Witkin, and M. Kass, "Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion," *AI Journal* **36**, pp. 911-123 (1988).

Figure 6: Context-Based Feature Extraction.