The Representation Space Paradigm of Concurrent Evolving Object Descriptions

Aaron F. Bobick, *Member, IEEE*, and Robert C. Bolles, *Member, IEEE*

Abstract— **A representation paradigm for instantiating and refining multiple, concurrent descriptions of an object from a sequence of imagery is presented. Tbis paradigm is designed to be used by the perception system of an autonomous robot that 1) needs to describe many types of objects, 2) initially detects objects at a distance and gradually acquires higher resolution data, and 3) continuously collects sensory input. We argue that multiple, concurrent descriptions of an object are necessary because different perceptual tasks are best performed using different representations and because different types of descriptions require different quality data to support their computation. Since the data change significantly over time, the paradigm supports the evolution of descriptions, progressing from crude 2-D "blob" descriptions to complete semantic models, such as bush, rock, and tree. To control this accumulation of new descriptions, we introduce the idea of *representation space*. Representation space is a lattice of representations that specifies the order in which they should be considered for describing an object. Each of the representations in the lattice is associated with an object only after the object has been described multiple times in the representation and the parameters of the representation have been judged to be "stable." We define stability in a statistical sense, enhanced by a set of explanations describing valid reasons for deviations from expected measurements. These explanations may draw on many types of knowledge, including the pbysics of the sensor, the performance of the segmentation procedure, and the reliability of the matching technique. To illustrate the power of these ideas, we have implemented a system, which we call TraX, that constructs and refines models of outdoor objects detected in sequences of range data.**

Index Terms—**Autonomous navigation; object representation; scale space.**

I. Introduction

MUCH OF COMPUTER vision research is directed at the problem of constructing computational descriptions of the world. To that end, many representations—description languages—have been devised to describe different types of objects and support different types of tasks (e.g., see [1], [10], [13]). In addition, there is an extensive body of research on filtering techniques for incrementally refining object description parameters as new sensory data are acquired. However, little research has been devoted to the coordination of multiple, concurrent descriptions of objects, particularly when the descriptions are to be refined over time. In this paper, we present a representation paradigm that supports the instantiation, accumulation, and refinement of significantly different descriptions of an object.

The goal of constructing a multiplicity of descriptions of an object is motivated by the following two observations: First, different objects and different tasks require different representations. A description language well suited for describing the shape of vegetation may be poorly suited to describing the shape of a hippopotamus. Second, as the quality of sensory data changes, the types of representations that can be supported change. The initial description of a distant object may be as simple as a bounding sphere, whereas a fully developed model, built from high resolution data, may be a complex structure of parts. It is premature to try to compute a multipart description of an object that spans only a few pixels in an image.

The motivation for our research is the development of a perception system for an autonomous robot. One of our primary goals for such a system is for it to construct a reliable model of the environment that is complete enough to support such tasks as route planning, obstacle detection, and landmark recognition. This need to support a wide range of tasks requires the perception system to compute a rich set of descriptions. In addition, within the domain of autonomous navigation, the availability of new and improved data arises naturally; approaching an object yields better resolution data, and repeated observations from different directions provides increasing amounts of shape information. To provide an intuition as to the desired performance of such a perception system and to motivate the use of multiple, concurrent descriptions, consider the following example of an autonomous system, constructing a map of its environment as it moves along.

Assume that a robot vehicle using range imagery initially detects a small object at a distance of 20 m (the obstacle is actually a thin thistle bush). At that range, the system cannot be certain whether the object is a real obstacle or an artifact of the detection process; confirmation from the analysis of subsequent images is required. By analyzing three or four new images of the scene, the program determines that the object is real and then formally enters the object into the robot's model of the environment. Poor sensor resolution, however, permits only a crude estimate of the object's size and position. As the vehicle continues to approach the object, the increased resolution allows the robot to specify the size and position more precisely; again, agreement between estimates from one image to the next provides a high degree of confidence in these

Manuscript received November 1, 1990; revised May 6, 1991. This work was supported by DARPA contracts MDA 903-86-C-0084 and DACA 76-85-C-0004.

A. F. Bobick was with the Artificial Intelligence Center, SRI International, Menlo Park, CA 94025. He is now with the Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

R. C. Bolles is with the Artificial Intelligence Center, SRI International, Menlo Park, CA 94025.

IEEE Log Number 9102681.

0162-8828/92$03.00 © 1992 IEEE

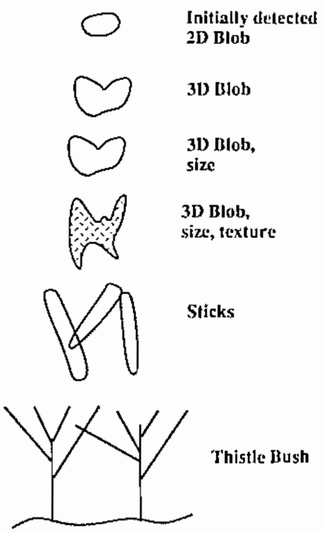


Fig. 1. Evolution of the description of an object. As more information becomes available, the parameters of previously instantiated descriptions are refined, and as more descriptions can be computed reliably, the model of the object is expanded to include these new descriptions.

estimates. As the vehicle gets closer yet, the program detects and builds descriptions of individual parts of the object. It detects and describes four stick-like parts that correspond to the stem and branches of the thistle. When these parts have been confirmed over several images, they are added to the object's model, and refinement procedures are instantiated to update their shape and location estimates over time. Finally, since the descriptions of the object's parts match those of thistle bushes, which are expected in the area, the robot classifies the object as a thistle bush, and adds this semantic description to the object's model. This cumulative description process is shown in Fig. 1.

If, during the analysis that produces these descriptions, the bush is not detected in an image, the program tries to explain why the bush was not detected instead of assuming that it disappeared. Perhaps the bush is out of the sensor's field of view, is occluded by another object, or was missed by the low-level segmentation process. Incorporating such an explanation subsystem into the description evaluation process improves performance by successfully accounting for infrequent, yet expected, situations. To identify the possible causes of loss of continuity, the explanation subsystem invokes a wide variety of heuristics designed to match the characteristics of particular sensory and processing stages.

As a perception system of the type we are describing computes and validates more descriptions of the objects in a scene, it is able to provide better responses to requests from other vehicle modules, such as the route planner or the landmark recognizer. For example, if the perception system has only computed and validated the crude 3-D description of an object, its response to a question about possible obstacles in front of the vehicle would only consist of a description of the object's approximate size and location. Not knowing the identity of the object, the planner would have to select a route that avoids the object. If, on the other hand, the perception module has identified the object as a thistle, the planner has more options, including running over the object, if there is no convenient clear path around it.

In this representation paradigm, the system only compares two object descriptions in the context of a specific task. The key question is the following: "Which description is better for answering the particular question?" and not "Which description is intrinsically better?" Thus, an occupancy grid may be the best description for answering questions about the empty space around an object; a viewer-centered description may be best for tracking an object from image to image; a generalized-cylinder model may be the best for predicting the appearance of an object from another point of view. The system employs several representations as equal partners in its description of the object.

In the remainder of this paper, we describe a representational framework for a vision system that maintains multiple, concurrent descriptions of objects. The representations used to form the descriptions are designed to model different types of objects, to support different types of inferences, and to require different specificity and accuracy of data to warrant their computation. We embed this framework within a system that incrementally constructs object descriptions over time such that the complete description of an object evolves. We make use of temporal stability to assess the validity of computed descriptions. In the final section of the paper, we discuss some the difficult questions that are raised by employing such a representational scheme.

Throughout this paper, we illustrate our ideas with results from the TraX system, which is an implemented system that constructs and refines models of outdoor objects, such as bushes, trees, and rocks, detected in sequences of range data. With regards to the implementation of TraX, we make two observations: First, we do not mean to imply that the particular set of representations presented is adequate to describe the entire outdoor world. In fact, our research is designed to allow the seamless introduction of additional representations as is necessary; additional representations are needed as new object types or new tasks are considered. Second, the particular components assembled for addressing the autonomous navigation task are not of primary importance; they were constrained by the sensory data available and the objects of interest. Rather, these components are used to illustrate the importance of incorporating a detailed understanding of the sensors and the processing algorithms into the multirepresentational framework; this understanding is critical to successfully choosing available representations and exploiting computed descriptions to perform necessary tasks.

II. A Space of Representations

In a multiple representation system, should all the representations be used to describe all the known objects all of the time? We argue that the answer to this question is "no" for two reasons: First, the resolution of the data may only support simple models. Not only would computing a more complex structural description be a waste of computational resources, but the model produced would be erroneous, possibly leading to false conclusions on the part of the perception system.

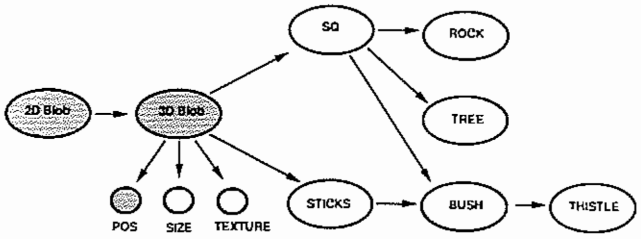
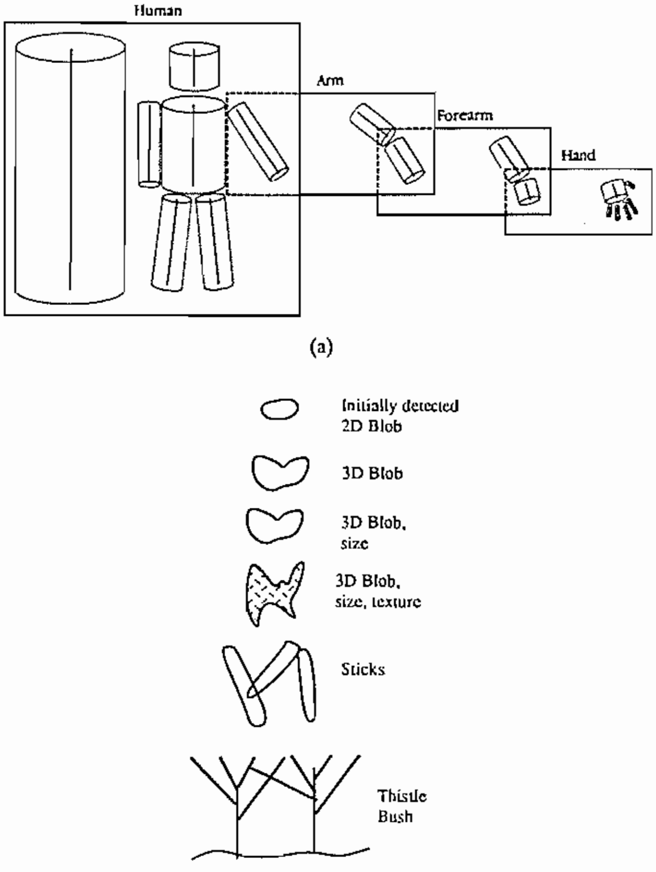


Fig. 2. Representation space for the TraX system. The shaded nodes represent components of the representation in use. A new node can be shaded only if one of its connecting nodes is shaded and the stability conditions necessary for its acceptance have been met.



(a)

(b)

Fig. 3. Different representation schemes: (a) Hierarchical representation; (b) representation space. In representation space, descriptions vary in the types of representations used. Additional information not only causes accuracy to improve but also allows an object description to contain different types of information. ((a) reprinted from [10]).

Second, the diversity of objects in the world is such that some objects are best described using one set of representations, whereas others are best characterized by another. It is unreasonable to expect a single representation to be appropriate for all objects in the outdoor world; this is especially true for high-level representations such as generalized cylinders [1], superquadrics [16], or geometric solids [17], [13]. In the domain of autonomous navigation, a building might be well represented by geometric solids, whereas more irregularly shaped objects, such as trees and bushes, would require quite different representations.

To capture the natural progression of representations supported by better data and to cover the diversity of objects in the world, we have introduced a partially ordered set—a lattice—of representations, which we call *representation space*. The importance of having a lattice is that it focuses the perception system on the most appropriate representations for an object, given both the resolution of the data and the inherent properties of the object.

Fig. 2 shows the representation space used in the TraX system, which we implemented to explore the issues associated with a multiple representation system. We consider representation space to be composed of *fundamental representations* and *enhancements*. Each fundamental representation reflects a qualitatively distinct representation, whereas an enhancement corresponds to the addition of a few parameters to a fundamental representation.[1] In the diagram, each large node corresponds to a fundamental representation, and each small node corresponds to an enhancement. As indicated, fundamental representations available in our TraX system include 2-D blobs, 3-D blobs, superquadrics (SQ), sticks (a 3-D parts representation described later), and several semantically based representations including bush and tree.

Representation space is similar to scale space [19] in that the representation of an object is not restricted to any one level of description; different levels of specificity are possible. Unlike scale space, however, and unlike hierarchical representations, e.g., [10], [12], [4], representation space is not homogeneous. For example, Marr and Nishihara [10] propose using generalized cylinders of many scales to achieve a representation that spans data of different resolutions. Although the description of an object improves as more detailed information is acquired, there is no change in the type of inferences the representation can support. Only the size and number of primitives and the corresponding level of accuracy improves. In representation space, however, a change in representation often implies the ability to assert new properties about an object. These two approaches are schematically contrasted in Fig. 3.

One of the implications of representation space is that as new data are processed, the description of an object can be modified in one of three ways. First, the parameters of the active components of the representation can be updated. We refer to this process as *refinement*; refinement procedures use standard filtering techniques and are similar to algorithms used by others to reduce parameter uncertainty [2], [11], [18], [5]. The second type of change is the activation of a parameter or property attached to an active representation. For example, the active representation indicated in Fig. 2 could be expanded by activating the TEXTURE node under 3D–BLOB. This type of modification is referred to as *enhancement*; the representation is enhanced by the addition of a new parameter. The final type of update is *augmentation*; in Fig. 2, this would correspond to activating either the SQ (superquadric) or the STICK fundamental representation. The augmentation of a representation for an object means that the object can be

[1] We recognize that there is no formal distinction between levels and parameters. However, the intuition that there are several qualitatively different representations, each of which can be enhanced by the addition of a few parameters, is strong, and we have found the distinction useful.

described in a completely new vocabulary. As a collection, the methods of modifying the description of an object are designed to combine well-known quantitative techniques for integrating information with a more qualitative approach that permits the nature of a representation to change over time.

Arcs in the representation space diagram indicate ways that the description of an object can be extended, that is, they provide the control structure for the accumulation of descriptions. A new node in representation space can become active, indicating that the corresponding representation is active for a given object, only if one of its predecessor nodes is active. By shading nodes in this diagram, we indicate the active components for a particular object. For example, in Fig. 2, the large shaded node labeled 3D–BLOB indicates that a reliable 3-D blob description has been computed for the object. The small shaded nodes labeled SIZE and POS reflect the fact that the size and position of the blob are known.[2] Thus, for this particular object, the TEXTURE, SQ, and STICKS nodes can be activated the next time data for this object is analyzed.

Note that the arcs in representation space do not imply computational dependency. For example, the algorithms in the TraX system for computing a superquadric model of an object are independent of those for computing a 3D–BLOB description. This differs from typical level-of-abstraction hierarchies, where each new description is computed from the previous level representation; in a typical sequence, lines are computed from edges, planar facets from lines, volumetric primitives from lines, etc. [8]. Such chaining of representations leads to the compounding of processing errors. In contrast, the different levels of representation space can be used to check the validity of a computed description. If the 3D–BLOB predicted by operations performed on the superquadric model is not similar to the blob computed directly from the data, then the system would have evidence that at least one of its descriptions is not valid. Although we have not yet explored this issue in detail, we would hope to make use of the independent computation of representations to increase the overall robustness of the system.

In addition, it is important to realize that when a new representation is invoked to compute a description for an object, the previous descriptions are not discarded. They are retained because they may be the best representation to answer a task-related question, even if they are at a reduced level of specificity or accuracy. Examples of employing multiple levels of description for accomplishing different perceptual tasks are included in the next section.

To underscore the point that the construction of a description of an object is a cumulative process, consider the conceptual graph in Fig. 4. This graph is intended to reflect the utility of representation space. The abscissa indicates the amount of processed sensory input, which in the case of an autonomous robot is monotonically related to time. The ordinate indicates the "power" of the description of an object as constructed by the system. As more data are processed, the description of an object becomes more precise and, thus, more "powerful." The large steps indicate augmentation, where a new level of representation has been activated that supports many new inferences about the object. The small steps reflect enhancement, where new parameters have been added to a current description. Finally, the increasing slope of the tops of the steps indicates refinement, which is the improvement in the accuracy of the current representation.

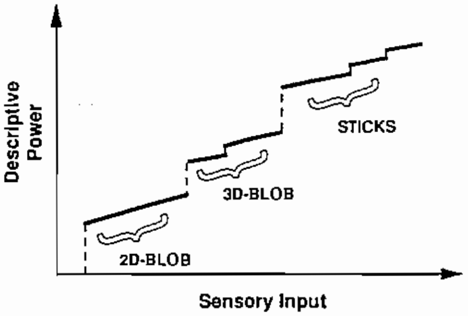


Fig. 4. Conceptual graph demonstrating the utility of representation space. The abscissa indicates increased sensory input. The ordinate indicates the "power" of the description of an object as constructed by the system. The large steps indicate *augmentation*, where a new fundamental representation has been activated and that new representation supports many new inferences about the object. The small steps reflect *enhancement*, where new parameters have been added to a current description. Finally, the increasing slope of the tops of the steps indicates *refinement*, which is the improvement in the accuracy of the current representation.

III. Stability and Validity

Representation space controls the order in which representations are explored for describing a particular object. How does the system decide that a computed description is *valid* and, therefore, should be added to the object's model? In this context, we use the term valid to mean that the description correctly characterizes some aspect of the object, as opposed to being a transient artifact of the processing.[3] To address the question of validity, we must consider the causes of artifacts.

Artifacts can arise for several reasons. Computing a description of an object using an inappropriate representation can easily lead to a model that does not reflect any intrinsic property of the object; for example, a stick description of a boulder is mostly determined by idiosyncracies of the stick fitting algorithm. In addition, artifacts can arise because of rare, yet expected, events that violate the assumptions embodied in the processing; for example, accidental alignment can make two objects appear to be one larger object. Finally, artifacts can occur because of unmodeled errors; for example, a segmentation algorithm can hallucinate an object from an unlikely variation in the data. The ability to determine when a description is valid is important for any perception system; it is critical in a multirepresentational framework in which many descriptions are tried, but only a few characterize an object well.

Our current approach to assessing the validity of a computed description relies on an analysis of temporal stability. We do

[2] For this discussion, we are ignoring the issue of uncertainty in the estimate of a parameter. In actuality, once the measurement of a parameter is determined to be relatively stable, we use Kalman filtering techniques to update the value of the parameter and maintain an explicit estimate of the uncertainty of the value.

[3] Our use of stability and validity is closely related to the "principle of stability" in [7].

this by tracking an object over time, computing a new description of it in each image, and then analyzing the sequence of these independently computed descriptions. If the descriptions are similar over a period of time, we compute a composite description, validate it as a real entity, and add its description to the model of the scene. For example, if a particular stick description is computed repeatedly for a part of an object, we assume that the consistency across independently computed descriptions is due to a real structural property of the part, and therefore, the stick description of the part is added to the model of the object.

Given this basic strategy, there are two key phrases that need to be functionally defined in order to convert it into an algorithm: "similar descriptions" and "over a period of time." To define these terms, we ideally would like to rely strictly on strong models of components of the perception system, such as the physics of the sensor, its noise characteristics, and the characterizations of the image analysis techniques. However, in practice, these models are not adequate to completely predict the behavior of the system. As a result, we use these explicit models when they are available and, when necessary, augment them with statistical, empirically determined models. For example, we predict where we expect to see a previously detected object and how large it will be from a combination of three strong models: a model of the physics of the range sensor, a model of its scanning geometry, and a model of the vehicle's motion based on inertial navigation data or land navigation data. However, to predict how frequently an object might be missed by our low-level segmentation procedures, we built a simple statistical model by applying the procedures to hundreds of images and accumulating failure statistics.

Deep and Shallow Models and Explanations: Jain and Binford [9] use the term "shallow" to refer to statistically derived models; we will thus use the term "deep" to refer to models derived from known physical systems. These deep models, such as the model of the range sensor's scanning technique, can support quite precise predictions and can be embedded in algorithms that cover a wide range of tasks. For example, we use the scanning model of the sensor in conjunction with high-frequency inertial navigation data (i.e., a set of measurements for each image scan line) to compensate for the bouncing of the vehicle during the 0.4 s required to gather a range image using the Environmental Research Institute of Michigan range sensor. This process not only corrects for bouncing but also corrects for the relativistic effect that causes vertical telephone poles to bend in the imagery because the vehicle is significantly closer to the pole when it measures the bottom of the pole than when it measures the top. Deep models are robust in the sense that they always contribute an accurate characterization of the processing system.

Shallow models, however, such as our simple statistical model of the failure frequency of our segmentation procedures, are always suspect. A commonly cited reason for their restricted utility is the difficulty of ensuring that the training set adequately covers the range of expected scenes [6]. A more serious difficulty with shallow models is that situations arise in which the application of those models is inappropriate. As mentioned, we use the empirically determined probability of failure of a segmentation algorithm to determine the number of times an object should be detected in a sequence of imagery before being declared valid. With our current segmentation procedures, this threshold is set at three. However, there are many reasons other than the failure of the segmentation procedure for an object to be undetected in a given image and, furthermore, these situations are predictable from additional system component models. Thus, for robust performance, the use of shallow models must be tempered by *explanations*, which are here defined to be understood situations that cause shallow models to be inappropriate. For example, the decision about whether an object has left the field of view is based on the model of the sensor's scanning technique and the vehicle's location estimates; this is an example of an explanation based on a deep model. On the other hand, one of the common mistakes of our system occurs in our column-oriented analysis of the raw range image. For that problem, we only have an ad hoc description of the pathology of an error; in this case, a columnar plume suddenly appears in the shape of the object. If some object is not matched in a new image, and an apparently new object has appeared with a large columnar piece, the system explains the situation as being a possible error in processing.

The idea of a *possible* error introduces our last point about employing shallow models; decisions based on shallow models are always suspect and should be confirmed by further data. As such, it often becomes necessary for the system to maintain multiple, competing hypotheses about the state of an object. Continuing the example of the last paragraph, the system does not absolutely conclude that the newly shaped object is indeed a hallucination caused by a processing error. Rather, a wait-and-see attitude is adopted, and both possibilities are maintained; the processing of subsequent images resolves the ambiguity.

A. Stability in Blob Detection

The first representation used to describe an object is 2-D-BLOB. In the TraX system, the 2-D-BLOB description of an object consists of the range pixels corresponding to the object, as viewed in the most recently processed image. This representation is important not only because it is the first instantiation of a model for an object, and therefore endows existence to some object, but also because the actual pixels viewed in one image are the best description for matching that object in the next image of a sequence. The question of whether a 2-D-BLOB description is valid is really a question of whether the segmentation process correctly detected a real obstacle or whether it mistakenly isolated some pixels that are part of the ground. Robustly detecting obstacles and tracking these objects from image to image are critical in a system that integrates information over time to construct reliable models.

The segmentation procedure in the TraX system consists of classifying each pixel in each range image as ground or obstacle. This classification is made by applying a multistep procedure that first identifies regions in the image that are well fitted by planes. We next determine the consistency of these planes with an *a priori* digital terrain map (DTM), using orientation as the principal factor. The consistent planes are then extended to completely cover the gaps between them, essentially forming a new, local DTM. Any range pixels that are more than a certain distance above or below this new DTM are then marked as obstacles. Because the ground clearance of the Martin Marietta Autonomous Land Vehicle is about 6 in, we use that value as a threshold. Notice that because we make use of the temporal stability of the detected obstacles to validate the segmentation, we can set the threshold according to the specifications of the task instead of concentrating on the expected single image false alarm rate.

As mentioned earlier, we use a shallow model of obstacle detection to determine the best control strategy for assessing validity. Empirically, we have determined that if an object detected in three consecutive images, then there is a very high probability that it is a real object. In addition, once an object has been validated, it is unlikely to be missed twice consecutively. We implement this simple control strategy using a quasi-finite-state machine (QFSM). We use the term "quasi" because as an object moves through these states, the history of its traversal is recorded and can sometimes affect operations that occur outside the FSM control structure[4]

Fig. 5 shows a simplified portion of the QFSM used for the analysis and tracking of 2-D blobs. Notice that there are several ways to enter the **Initially-Detected** state, including being detected in the first image of the sequence, coming out from behind another object, and splitting off a previously detected object. The importance of making these paths explicit is that we can later use this information to help explain unexpected phenomena and to affect decisions made outside the control structure of this QFSM, such as deciding how to combine the models of two objects that are later decided to be only one object.

Once an object is **Initially-Detected**, we try to match that object in subsequent images. As an object is successfully matched, it moves into the **Stable** state; at this point, the object is considered to be "real," and attempts to extend the description are begun. If, however, after initial detection the object is no longer matched, the object quickly moves to the **Artifact** state, indicating that the detected obstacle is an artifact of some processing step and should be discarded.

Notice that at each state, there is a **missed-but-can-explain** transition. This type of arc represents a situation in which the object is not successfully located in an image in which it is expected, but there is an external explanation as to why not. Increasing the competence of the system requires recognizing these situations and incorporating explanations of them into the evaluation process. We currently have implemented the analysis required to support the following explanations:

- The object is no longer in the field of view of the sensor.
- The object is occluded by another known object.
- The object is a small, short blob far away; therefore, it can be easily missed.
- The object merged with another object to form a larger object.

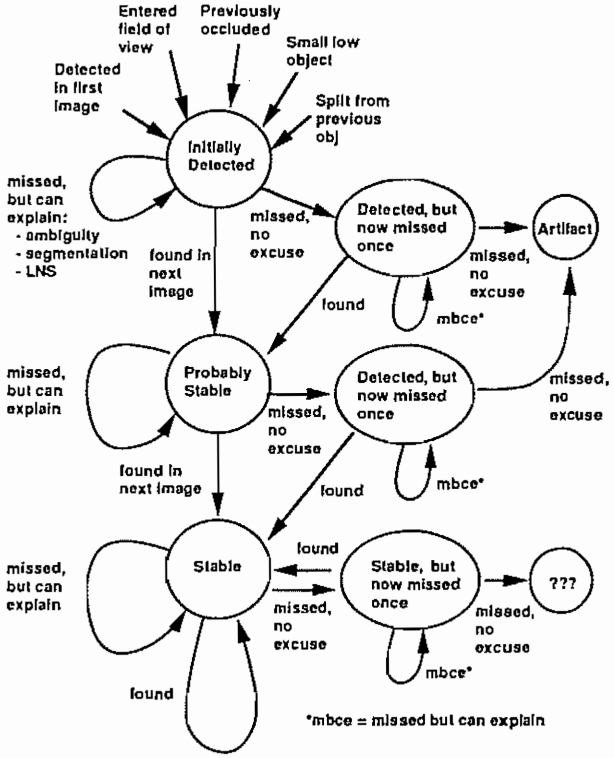


Fig. 5. Part of the finite-state machine for determining stability of 2-D–BLOBS.

- The object is unmatched because an error in the ambiguity interval assignment greatly changed the apparent characteristics of the object. (Ambiguity interval assignment is a preprocessing step that is necessary for determining the true range from a phase shift range image.)

Fig. 6 is a sequence of segmentation images produced by the single-image analysis. Object 1 (the short object on the right) is detected in all four images. Initially, this object is not matched in the fourth image because the object's shape has changed dramatically. However, the change is mostly characterized by the addition of a column of pixels in the last image. In the TraX system, a column-scanning algorithm is used to disambiguate the phase-encoded raw range signals; we refer to this processing as an ambiguity interval assignment. Therefore, a column-shaped change in the appearance of an object is symptomatic of a particular mistake, namely, an ambiguity interval assignment error. Thus, in this example, the program concludes that an ambiguity error has possibly occurred. The TraX system retains information about this error as indicated by the following portion of program output:

```
...
Starting view 157 ...
(OBJECT-TRACKER-MBCE < OT ID:22
   :GENERIC-OBJECT-4 >
      AMBIGUITY-INTERVAL-PROBLEM-WITH-BLOB
< TRAX-RANGE-BLOB R:25/157 >
   BLOB-HAS-THE-EXTRA-PIECE 1-TO-1 CASE-6)
...
Creating OT < OT ID:41 :GENERIC-OBJECT-23 >
   for region < TRAX-RANGE-BLOB R:25/157 >.
...
```

[4] We could implement the control structure using a true FSM by simply increasing the number of states. We choose not to do so because we would end up with many states that were qualitatively similar, obscuring the general structure.

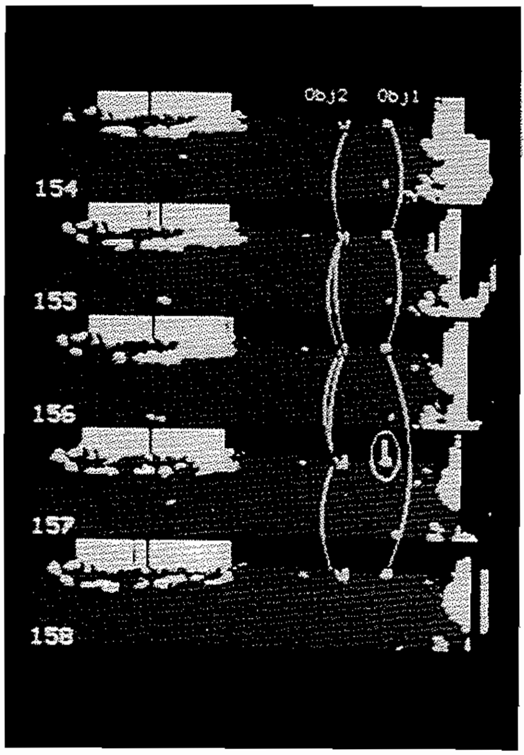


Fig. 6. Tracking detected obstacles from image to image.

This fragment indicates two things: First `:GENERIC-OBJECT-4` (referred to as object 1 in Fig. 6) was missed (not seen as expected) but could be explained (MBCE) by an ambiguity interval problem in the processing of blob 25 in image 157. Second, the system also starts a new object tracker (OT for `:GENERIC-OBJECT-23`) as a competing hypothesis that needs to be resolved in later processing. Because the original object `:GENERIC-OBJECT-4` was seen again in images 158 and 159 and because `:GENERIC-OBJECT-23` is not matched, the newly created object is quickly eliminated, with the explanation being that its creation was indeed an artifact.

Object 2 (the thistle bush to the left of object 1 in Fig. 6) is an example of a single object splitting (third image) and then merging again. In order to build a robust model of the environment, the program must be able to handle situations such as these. Again, the TraX system handles these ambiguities by generating competing hypotheses and resolving them with the processing of additional data. The density of these events in this short sequence is higher than usual, but they are typical of the events that occur in our analysis of hundreds of images.

In the future, we plan to expand the list of possible explanations. As we understand more of the fundamental properties of objects and more about the behavior of the analysis procedures, we can implement more explanations, increasing the competence of the system.

B. Stability in Integration

Once we have determined an object is real, we have a set of techniques for generating 3-D descriptions. The simplest uses a "3-D-BLOB" analysis that describes an object according to its position, size, and, potentially, surface texture. Initial 3-D analysis of a blob establishes an object-centered coordinate system and then computes a 3-D scene location for the object. The object-centered frame allows for the integration of information about the shape of the object to be decoupled from the compilation of information about the location of the object. The actual representation of this new 3-D-BLOB is an ellipsoid whose parameters of position and size are updated using standard Kalman filtering techniques. Critical to this integration is knowledge of the noise characteristics of the sensor.

When more precise range data are available, we can compute 3-D part models using one of two representations: superquadrics and "sticks." These representations support a class of inferences about objects that are not supported by the blob descriptions, namely, those requiring a structured shape description.

Superquadrics are well suited to describing shapes composed of compact parts [14], [16]. The technique we use to compute superquadric models of objects is a modified version of the algorithm described in [15]. We first compute a "minimal cost covering" of the range pixels by executing a coarse global search over the superquadric parameter space and then optimizing the model by gradient descent. We have found this technique to be adequate for modeling simple objects such as rocks but have not exercised the algorithm enough to evaluate it fully.

When objects are composed of thin pieces, as are fence posts and thistle bushes, the response of the range sensor tends to "fatten" the parts by generating mixed pixels along the sides. This blurring prevents the superquadric algorithm from finding the true stick-like description. To model these thin objects, we have designed special representations we call "sticks." By definition, sticks appear as one-pixel wide lines in range images. Thus, to compute a stick model of an object, we first thin the range image of the object and then compute a minimal covering in a manner analogous to superquadrics. The stick model representation is used in the bush example presented later in this paper.

Fig. 7 displays the results of applying the stick-fitting procedure to a detected object. Each model is computed independently making no use of the previous solution. Note that most of the resulting models capture some structure of the bush. However, except for the last one, none captures all of the structure. The principal problem associated with these fitting techniques is the lack of data to constrain the models. As a result, there are often many descriptions that characterize an object equally well. As with obstacle detection, we rely on processes monitoring the stability of computed descriptions to filter out those that are not valid.

To integrate stick descriptions over time, we employ a method similar to that previously discussed for tracking of 2-D-BLOB's. In this case, however, new sticks computed from the data are matched to model sticks that are being refined with each image. Model refinement requires three stages: First, model sticks that are matched by new sticks are reinforced in terms of their stability, and their parameters are updated

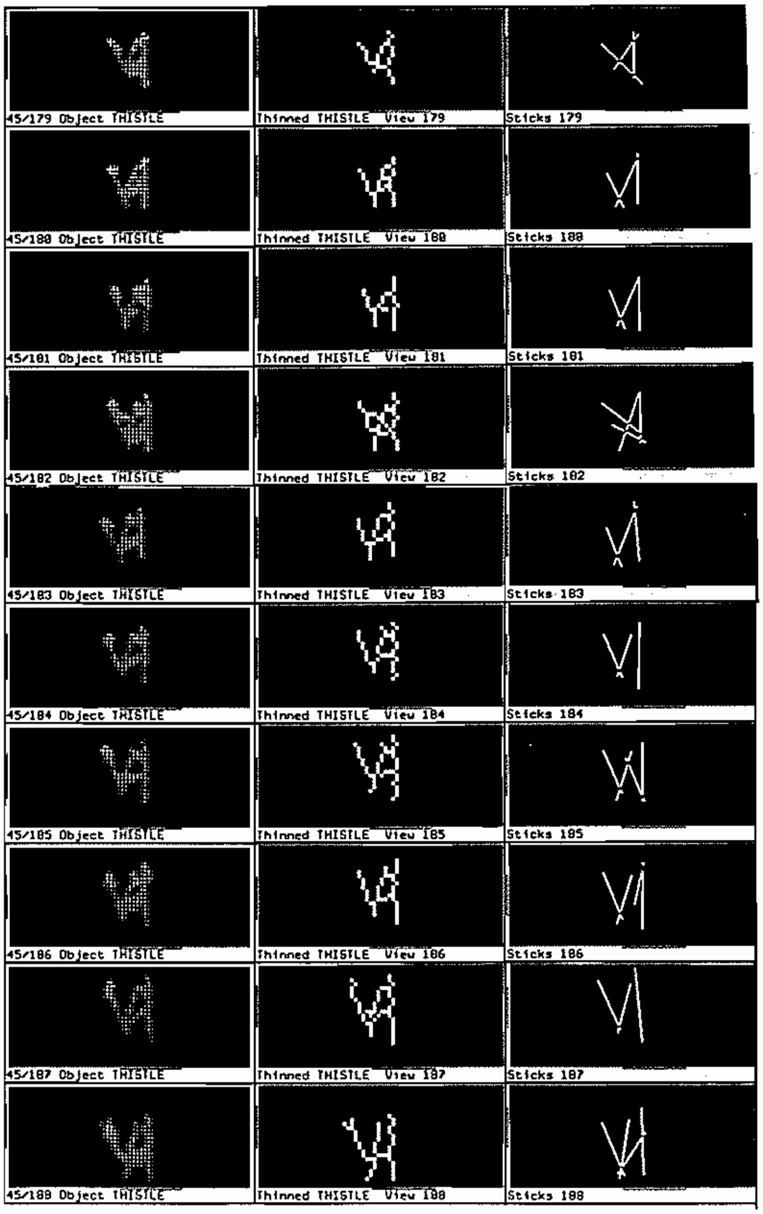


Fig. 7. Computing stick descriptions of a thistle bush. The column on the left displays the silhouette of the object as determined by the obstacle detection procedure; the middle column is the thinned version of these objects. The right column displays the best "stick" model of the thinned bush as computed independently for that one image. Note that some of the sticks are quite robust, such as the vertical stick on the right. Others are less stable, whereas some are artifacts. Although the fitting technique can be improved, our goal is to use temporal stability to compute a more robust model.

using standard Kalman filtering techniques; the state variables estimated are the endpoints of each stick [2], and we use our model of the sensor and its noise characteristics to estimate the variance of the new measurements. Second, unmatched new sticks cause the formation of model sticks that are initialized from the data; these sticks are searched for in subsequent images. Finally, a stick that was initially detected but not matched again is eventually discarded as an artifact of the stick-fitting procedure, unless there is an explanation as to why the stick should not be matched. Currently, we include only one explanation that allows an unmatched stick to remain as a viable part of the representation. The vehicle has backed away from the object, and the individual stick may no longer be detectable by the sensor.

Fig. 8 shows an example of the stability analysis applied to sticks. The stick description computed independently using the single range image as input is on the left. The set of stable sticks tracked over time is on the right. A new stick is added to the model on the right only after it has been deemed stable. Note that the stable description converges to (what is known to be) an accurate model of the bush.

IV. Hard Problems

A. *Quasi-stability*

In our approach to temporal integration, temporal stability of a description is the primary indicator of reliability. The basic

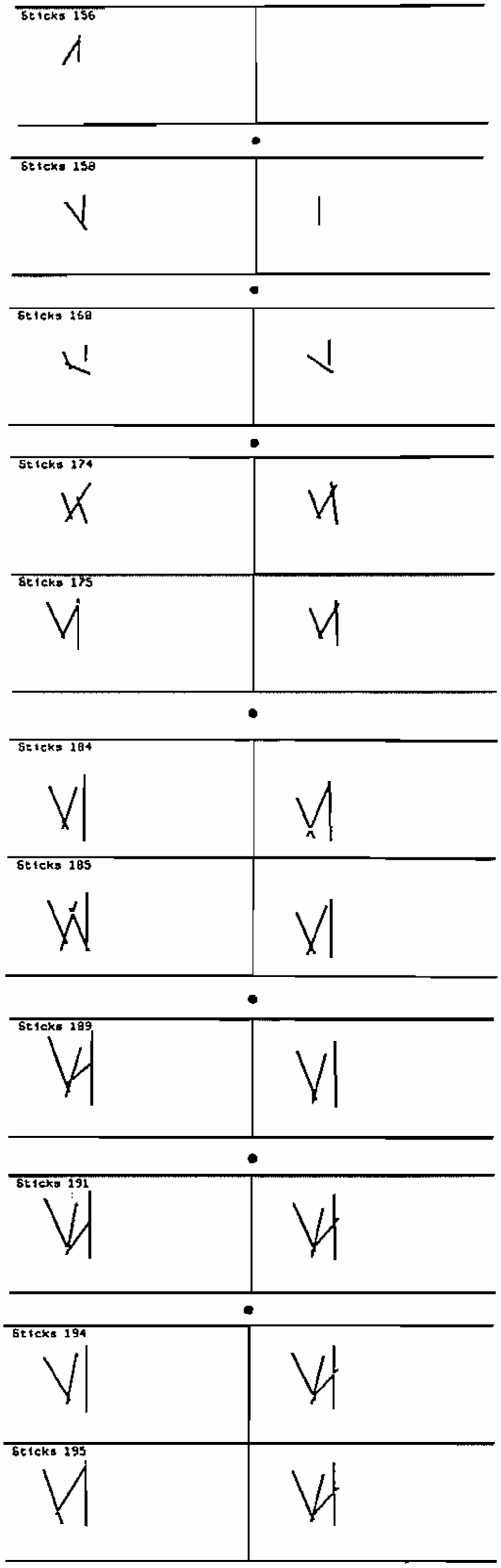


Fig. 8. Sticks computed independently (left column) and tracked over time (right column). As a stick becomes stable, it is added to the model on the right.

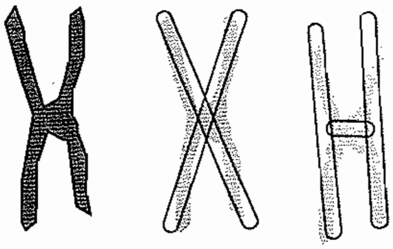


Fig. 9. Thin object equally well described as a two-stick 'X' or as a three-stick 'H.' Stability analysis cannot be used to resolve this ambiguity.

assumption of this strategy is that for each appropriate representation, there is a unique, correct description of an object and that commonality across independently computed descriptions reflects valid aspects of those descriptions. However, the appropriateness of this assumption depends on the match between the objects in the domain and the representations. Consider, for example, the image of an object shown in Fig. 9. In this case, describing the object as a two-stick 'X' or as a three-stick 'H' is equally correct. In addition, stability cannot disambiguate between these models because they may each occur periodically; such an occurrence leads to two competing stick models, each of which is "quasi-stable."

Although we do not have a complete solution to this problem, we can avoid most of these difficulties in the TraX system by keeping track of competing hypotheses. Thus, the system could maintain two or more descriptions of an object in one representation and clearly mark them as alternatives. If asked to assert or predict a property of this object (such as an appearance from some viewing direction), the system would have to decide which description or which combination of descriptions it should use, just as it does now when answering a task-level request about an object that has multiple descriptions derived from different representations. Although this remedy may be adequate for many problems, it is clearly unsatisfactory for situations in which there are many quasi-stable descriptions of an object.

A related problem arises when the assumption that temporal stability reflects validity is violated. If the data on which an inappropriate model is constructed do not change over time, then that inappropriate model will be deemed valid. Our current explanation strategy provides a mechanism for explaining the lack of confirmation of a correct interpretation but does not handle this inverse situation.

B. Dynamic Worlds

In the TraX system, the scene is assumed to be static; other than the vehicle, objects do not change their location, orientation, or shape. What are the issues in extending our approach to a dynamic environment?

One response is to simply model the dynamics of the environment. In this case, variables such as velocity and acceleration become additional parameters of the representation; aside from incorporating these new variables into the prediction mechanisms, the approach to temporal integration remains the same. In this case, however, stability becomes much harder to assess. If an object is moving (e.g., a rotating windmill), how does one determine that the shape description computed is stable, implying that the description is valid? Presumably, an understanding of the dynamics would need to be included in the model itself.

Another issue raised by a dynamic environment concerns the matching of known objects to objects detected in the data. If an object can change in location, orientation, and shape, how does one determine correspondence? Without digressing into a philosophical discussion of ontology or Lincoln's axe,[5] we must consider how to ensure that one is integrating information about the *same* object. One insight into this problem is derived from the work on motion analysis developed by one of the authors [3], where the temporal sampling rate is high enough that all transitions over time are smooth with respect to the data. Thus, for example, tracking the movement of an object, such as a person's arm, is simplified because the object can be easily tracked from frame to frame. That approach requires that the data sampling rate be high enough to smoothly sample the dynamics of the domain.

C. Explanations and Hallucinations

In this paper, we have made the claim that increasing the number of explanations that the system can invoke to explain why the actual sensory data deviated from predicted data increases the overall competence of the system. The intuitive argument supporting this claim is clear; if the number of important events that the program can diagnose is greater, the less likely the data are to confuse the model construction process. However, one must be aware that if the system has enough explanations, then the system can find an explanation for anything. As an extreme example, suppose the explanation "The sensor is completely broken and the incoming signal is independent of the world." is part of the knowledge base of the system. Then, any data may be explained by such a statement, and no useful modeling occurs.

To avoid such confusion, we have to resort to the idea of *best* explanation, where best is defined as most likely according to some *a priori* model of the world. This approach is the same as that adopted by researchers employing minimal encoding strategies to select the best description of a scene; examples include segmentation and parts descriptions [16]. One difficulty with this stratgey is that it requires the assignment of prior probabilities to low-probability events (e.g., the sensor has completely failed); these types of prior probabilities are usually suspect.

To date, we have avoided addressing this problem by placing stringent preconditions on the invocation of most explanations. These conditions are strong enough that if they are satisfied, we are willing to state categorically that the explanation is appropriate. In the few instances of shallow model explanations where strong preconditions do not exist, the requirement that the weak conditions remain true over an extended period of time prevents the explanations from becoming too widely applied.

V. Summary

We have described a new representation paradigm that supports concurrent evolving descriptions of an object. Our rationale for developing this paradigm is as follows:

- Multiple, concurrent descriptions are required for two reasons: 1) to describe the wide variety of objects that occur in complex domains, such as the outdoor world and 2) to efficiently support the inferences required by a collection of task modules, including object tracking, path planning, obstacle detection, and landmark recognition.
- Not all representations are appropriate for every detected object. Sometimes, the data are not sufficient to support the representations, and sometimes the representations are simply not appropriate for the object, such as a fractal model of a hippopotamus. Therefore, to restrict the application of representations to appropriate objects, we introduce the idea of a representation space, which imposes a partial ordering on the set of available representations.
- For applications in which a continuous stream of data is available, the descriptions of an object can evolve in two ways. First, the parameters of a representation can be refined by filtering techniques as new data are acquired, and second, if the data improve over time, new descriptions can be added when they are supported by the data.
- Temporal consistency across independently computed descriptions of an object is a strong indication of the validity of the descriptions. If the same description is computed from several images, there is a high probability that the description captures a real structural aspect of the object.
- Since there are many reasons for a description to change from one image to the next, the idea of temporal stability can be significantly enhanced by the addition of explanations that account for the problems and special cases that invariably arise in the processing of real imagery. The sources of explanations range from deep models, such as the physics of the sensor, to shallow models, such as the probability that a low-level procedure makes a mistake.

The ability to change an object's description incrementally and to build a temporally persistent yet consistent model of the environment is crucial in autonomous navigation tasks; objects are viewed many times from different viewpoints and with different resolutions. By continually updating the objects' descriptions, a robot is in a position to base its decisions on the most current information at all levels.

Acknowledgment

The authors gratefully acknowledge the comments of the SRI vision group and the assistance of T. Strat, A. Hanson, S. Barnard, H. Wolf, and L. Quam. The range data were gathered at Martin Marietta, Denver, CO, with the assistance of S. Seida, J. Allison, T. Dunlay, and L. Haber.

References

[1] G. J. Agin and T. O. Binford, "Computer description of curved objects," in *Proc. Third IJCAI* (Stanford, CA), Aug. 1973, pp. 629–640.

[2] N. Ayache and O. D. Faugeras, "Maintaining representations of the environment of a mobile robot," in *Proc. Int. Symp. Robotics Res.* (Santa Cruz, CA), 1987.

[5] Old joke: A farmer displays an axe over his mantle with a sign that reads "Abe Lincoln's Axe." When a skeptical visitor enquires about its authenticity the farmer replies: "Yup, it sure is Lincoln's axe. I've had to replace the handle twice and blade once, but it's old Abe's."

[3] H. Baker and R. Bolles, "Generalizing epipolar-plane image analysis on the spatiotemporal surface," *Int. J. Comput. Vision*, vol. 3, pp. 33–49, 1989.

[4] R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *J. Artificial Intell.*, vol. 17, pp. 285–348, 1981.

[5] J. L. Crowley, [1989], "World modeling and position estimation for a mobile robot using ultrasonic ranging," in *Proc. Conf. Robotics Automat.* (Scottsdale, AZ), 1989, pp. 674–680.

[6] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[7] M. A. Fischler and R. C. Bolles, "Perceptual organization and the curve partitioning problem," in*Proc. Eighth IJCAI* (Karlsruhe, West Germany), pp. 1014–1018.

[8] A. R. Hanson and E. M. Riseman, "The VISIONS image understanding system," in *Advances in Computer Vision* (C. Brown, Ed.). Erlbaum, 1987.

[9] R. Jain and T. Binford, "Ignorance, myopia, and naivete in computer vision systems," *CVGIP*, vol. 53, no. 1, pp. 112–117, 1991.

[10] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," in *Proc. R. Soc. Lond. B*, 1978, pp. 269–294, vol. 200.

[11] L. Matthies and T. Kanade, "The cycle of uncertainty and constraint in robot perception," in *Proc. Int. Symp. Robotics Res.* (Santa Cruz, CA), 1987.

[12] H. K. Nishihara, "Intensity, visible-surface, and volumetric representations," *J. Artificial Intell.*, vol. 17, pp. 265–284, 1981.

[13] M. Oshima and Y. Shirai, "A scene description method using three-dimensional information," *Patt. Recogn.*, vol. 11, pp. 9–17, 1978.

[14] A. P. Pentland, "Perceptual organization and representation of natural form," *J. Artificial Intell.*, vol. 28, pp. 293–331, 1986.

[15] ——, "Recognition by parts," SRI Tech. Rep. 406, 1986.

[16] A. P. Pentland and R. C. Bolles, "Learning and recognition in natural Environments," in *Proc. SDF Benchmark Symp. Robotics Res.*, 1988.

[17] R. J. Popplestone, C. M. Brown, A. P. Ambler, and G. F. Crawford, "Forming models of plane-and-cylinder-faceted bodies from light stripes," in*Proc. Fourth IJCAI* (Tbilisi, Georgia, USSR), 1975, pp. 664–668.

[18] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Proc. Int. Symp. Robotics Res.* (Santa Cruz, CA), 1987.

[19] A. Witkin, "Scale space filtering," in *Proc. Eighth IJCAI* (Karlsruhe, West Germany), 1983, pp. 1017–1022.

Aaron F. Bobick (M'90) was born in Queens, NY, in 1959. He received the B. S. degrees in mathematics and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1981 and the Ph.D. degree in cognitive science from MIT in 1987.

Until 1992, he was a Computer Scientist in the Artificial Intelligence Laboratory at SRI International, Menlo Park, CA. His research activities ranged from a dissertation in cognitive science about object characterization and perceptual inference to recent work on the fusion of shape from shading with stereo. He has also collaberated with the perception group at the David Sarnoff Research Center on motion vision research. He is currently an Assistant Professor with the Vision and Modeling Group of the Media Laboratory at MIT.

Dr. Bobick is the chairman of the Image Understanding Technical Group of the Optical Society of America (OSA) and is a member of AAAI.

Robert C. Bolles (M'80) received the B. S. degree from Yale University, New Haven, CT, with honors in mathematics in 1967, the M. S. E. E. from the University of Pennsylvania, Philadelphia, in 1968, and the Ph.D. degree in computer science from Stanford University, Stanford, CA, in 1977.

He is currently a Principal Scientist at SRI International, Menlo Park, CA, where he has performed computer vision research for 15 years. He has pioneered feature-based recognition techniques, automatic vision-programming techniques, and a spacio-temporal structure-from-motion technique known as Epipolar-Plane Image Analysis. He has written numerous papers in these areas and is currently co-principal investigator of SRI's image understanding projects.

Dr. Bolles is an editor of the *International Journal of Computer Vision*. He is also on the Board of the International Foundation of Robotics Research and is a member of the AAAI.