SRI International

LEARNING AND RECOGNITION IN NATURAL ENVIRONMENTS

Technical Note No. 421

June 5, 1987

By: Alex P. Pentland

Artificial Intelligence Center
Computer Science and Technology Division

Approved for public release; distribution unlimited

Support for this work was provided by the the National Science Foundation, Grant No. DCR-83-12766; by the Defense Advanced Research Projects Agency, Contract No. DACA76-85-C-0004; and by a grant by the Systems Development foundation.

SRI International®

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Learning And Recognition in Natural Environments

A. Pentland and R. Bolles

Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, California 94025

Abstract

We present a system for learning descriptions of objects, and for subsequently recognizing learned objects, that functions in outdoor, natural environments. We describe in detail two modes of functioning within this system: (1) unguided, bottom-up learning of object descriptions directly from image data; (2) top-down recognition of objects whose approximate position and structure are known by use of image-level matching. We then argue that the systems's performance in these two modes indicates that robust outdoor performance can be achieved within the structure of our vision system.

1 INTRODUCTION

DARPA's Autonomous Land Vehicle (ALV) project is intended to develop and demonstrate vision systems that can navigate in outdoor, natural environments. As we see it, the major challenges faced by this project are to develop (1) a general-purpose vocabulary of models that is sufficient to describe most of the important landmarks that the vehicle will encounter, (2) recognition techniques that will allow us to locate these landmarks from sensor data in a directed, top-down manner, and (3) learning techniques that will allow us to compute stable, accurate descriptions of interesting landmarks in an unguided bottom-up fashion.

If a machine vision system possessed these capabilities, it could then identify potentially useful landmarks, enter their descriptions into a database, and then, during a subsequent traversal of the same region, search for and recognize previously-seen landmarks. Thus the capacity to describe real-world scenes using a vocabulary of models that is recognizable from the data would make it possible to accumulate a comprehensive catalog of landmarks that can be recognized and then used to guide the vehicle. This, then, is the goal of the work described herein.

1.1 Approaches to Machine Vision

Most machine vision systems may be divided into a *prediction* phase and a *description* phase, as is illustrated by Figure (1). They may then be categorized by the type of representation used for recognition — that is, by the type of descriptions they match when establishing a correspondence between the predicted object appearance and the image description.

We identify four major types of representation, each corresponding to a certain level of abstraction. The lowest level of abstraction is the *image*; an array of numbers that represent either the sensor data or a continuous, topology-preserving transform of the sensor data.

This research was made possible by National Science Foundation, Grant No. DCR-83-12766, by Defense Advanced Research Projects Agency Contract No. DACA76-85-C-004, and by a grant from the Systems Development Foundation. We wish to especially thank Marty Fischler for his help, comments, and insight.

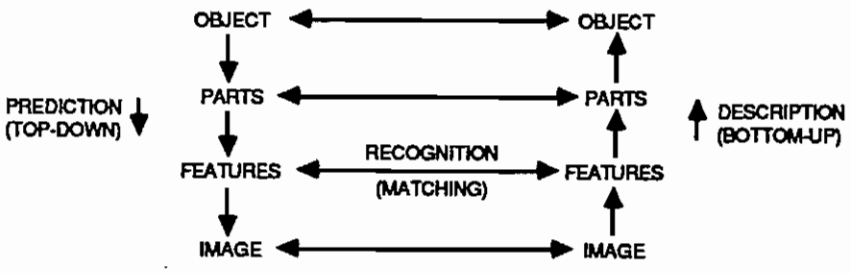


Figure 1: Possible levels of analysis for the learning-recognition process.

This level of representation retains the analog, nonsymbolic nature of the sensor data. Slightly more abstract is the *feature* level; a representation of image appearance in terms of such discrete elements as edges, corners, and circles. This level of representation is symbolic rather than analog.

At a higher level of abstraction is the representation of the viewed objects in terms of their component *parts*, by means of constructive solid geometry (CSG) [1], generalized cylinders [2], or the superquadrics-and-deformations representation employed in this paper [14]. This is the first level of representation that refers primarily to the three-dimensional world being viewed, rather than being tightly tied to the image. Finally, there is the representation of the scene in terms of *objects*, such as people, cars, and airplanes. At this level it is the *goals* of the vision system that become important, because what constitutes "an object" depends upon the viewer's purpose.

1.1.1 Recognizing Objects

The recognition task to be performed dictates the level of abstraction that is most appropriate. For instance, most industrial vision systems use the *feature* level of abstraction, skipping the part and image levels almost entirely. In such a system, the object to be recognized is first described in terms of the stable, easily found features, such as edges. This results in a wire-frame model, in which the "wires" of the model correspond to edges of the object.

We would then use this object model to form *predictions*, as illustrated by the left-hand side of Figure 1. For each viewpoint we would then project these wires into an image in order to form a *prediction* as to how these edges will appear when seen from that particular perspective. This prediction stage is normally accomplished beforehand, as a precompilation step.

To recognize the modeled object we then form a *description* of the image in terms of our chosen feature, as is depicted in the right-hand side of Figure 1. Typically, this means using an edge finder in order to locate long, straight edges. The actual step of object recognition, therefore, is reduced to establishing a correspondence between the features predicted relative to some viewpoint and the image description, as is shown by the arrows running between the prediction and description sides of Figure 1. This is accomplished by searching for a good match between some viewpoint's predicted edges and the edge description recovered from the image itself.

1.1.2 Learning Object Descriptions

The most important limitation of the foregoing industrial vision approach is that there is no way to learn the object description used for recognition *automatically*. Although relatively less important in most industrial applications this ability is crucial in the ALV domain.

As summarized on the right-hand side of Figure 1, the process of learning object descriptions is consists of computing a description — in terms of image, feature, or part representations — of interesting objects. For such a description to be useful, we must be able to compute this description is a totally bottom-up manner. This is why the task of learning object descriptions is the greatest hurdle faced by the ALV, since it requires that the program *automatically* segment the data, select the appropriate modeling primitives, and finally to instantiate the model's parameters.

1.2 Learning and Recognition in the ALV Domain

The levels of representation we choose depend upon the task to be performed. We must start, therefore, by listing the important tasks confronting our ALV. We believe that the following requirements are both necessary and sufficient, to ensure that potentially-useful landmarks will be identified, their descriptions learned, and these descriptions then used for recognition during subsequent traversals of the terrain.

- Obstacle detection — Deviations from "smooth" terrain that are large enough to impede the progress of the vehicle, or which are distinctive enought to be useful in navigation, must be detected.
- Description — Stable and accurate description of these obstacles must be constructed.
- Prediction: — The appearance of such obstacles, when seen from other viewpoints or when other sensors are used, must be predicted.
- Recognition — These predictions must be used to recognize and locate the known obstacles in new imagery.
- Refinement — Information based on the new imagery is used to update and refine object descriptions.

In the context of Figure 1, these tasks translate into the following procedures:

- Separate figure from ground in the image data to obtain an image-level description of the sensed obstacle.

- Compute an object description from this image description (proceed up the description side of Figure 1).
- Use this object description to predict either part, feature or image level descriptions (i.e., proceed down the prediction side of Figure 1).
- Establish a correspondence between the predicted appearance from some viewpoint and the (bottom-up) computed image description (i.e., connect between the description and prediction sides of Figure 1).
- Use this correspondence to update the object model (i.e., use the connection established between the prediction and description sides of Figure 1 to compute an improved object model).

With regards to Figure 1, then, there are only two remaining questions: (1) What level of representation is best for recognition? and (2) What level is best for learning descriptions?

The answer to the recognition question, we believe, is that *each* level — image, feature, and part — should contribute to the recognition process. Exactly how each will be utilized depends upon the precise nature of the context, sensor, and object description. We envision a flexible, opportunistic vision system built within this framework, one that can draw upon whatever information is available. For instance, if we know exactly what a landmark looks like and we have a good estimate of our position, then it is most efficient to match at the image level. On the other hand, if we know only that there is "a pole" out there, we are compelled to match at the part level — i.e., to find "a thing" that is approximately vertical, and is much longer than it is wide.

Our response to the learning problem, however, is that we must be able to form descriptions at the part level of representation so as to achieve stable and accurate learning of object descriptions. Such learning entails computing a description of the object that is sufficiently canonical that we can later use it for recognition. The fact that people can recognize objects and learn their descriptions strongly supports the view that there is some "natural," stable method for structuring object descriptions and that, moreover, people somehow recovere this "natural" structure from imagery and use it to support recognition and learning. We believe that this "natural" structuring of object descriptions is closely related to people's naive perceptual notion that objects have "parts." We shall argue, consequently, that most natural objects have a *part structure* that is recoverable from image data and can be used for recognition and learning. The specifics of the part-level representation employed in this paper were originally suggested by psychological evidence about people's notions of part structure [13,14].

This is not to say that feature and image levels do not both contribute to learning object descriptions; it is clear, after all, that the presence of T-junctions, parallel lines, and the like strongly constrain part structure [2-4,10-12]. Nevertheless, feature-level or image-level descriptions *alone* do not seem sufficient for generating descriptions that support general activities. There is, for instance, a substantial medical literature concerning patients whose spatial and feature-recognition abilities remain intact, but who are unable to recognize objects except in very special, highly constraining situations [34].

1.3 Outline of this Paper

In this paper we shall first describe a part representation that we believe facilitates the learning and recognition processes by providing the requisite descriptive power without involving a large number of parameters. Second, we shall describe a method for learning object descriptions, starting from ALV range data, that are based upon this part representation. Finally, we shall describe a method for recognizing previously learned objects by matching between predicted image appearance and measured image data.

With regard to the diagram in Figure (1), the techniques I describe herein represent two extremes among the possible approaches. The learning algorithm builds object models from the data in a completely unguided, bottom-up fashion. Moreover, it does this by direct image-to-part search, skipping the feature level entirely. The recognition algorithm predicts object's image appearance and matches at the image level, again skipping the feature level.

Although we have chosen to investigate only part-level representation and image-level matching, we anticipate that in a fully developed system *all* levels of representation would be present and that, especially in the recognition phase, there would be interactions at each of these levels. Our intent in choosing to explore these particular algorithms is to demonstrate that our general approach is sufficiently robust that it can also be successfully applied to both part-level and feature-level matching.

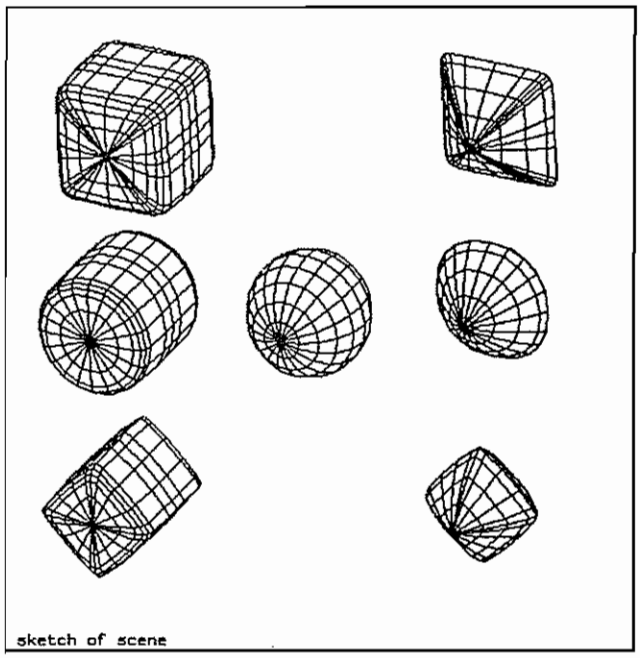
For instance, matching at the level of part descriptions — i.e., matching stored part-structure descriptions against a decomposition of the image into parts — is of particular importance for categorical and analogical reasoning tasks [16,31,35,36]. It is our contention that the ability to recover [reasonably] canonical part descriptions from image data makes it likely that our system can be used successfully for part-level matching. Because we can recognize objects without using feature-level information, we also consider it likely that our system's performance will improve when that information becomes available.

Another reason for choosing to investigate image-level matching is the similarity of range data to Marr's 2 1/2-D sketch: both are relatively dense measurements of the scene's intrinsic geometry. We believe that our range data techniques for learning descriptions and recognizing objects are transferable to situations where shape-from-x and depth-from-x methods, rather than a laser rangefinder, have provided us with image-like descriptions of a scene's intrinsic geometry.

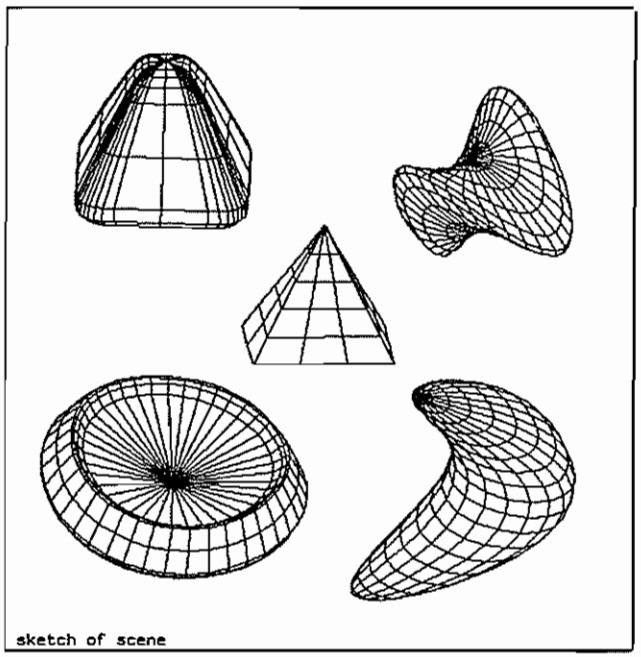
2 A PART REPRESENTATION

Many modern psychologists [10-12], as well as the psychologists of the Gestalt school, have argued that we conceive of the world in terms of *parts*, and that the first stages of human perception are primarily concerned with decomposing an image into these parts. This part-structure is seen as forming the building blocks for the rest of our perceptual interpretation.

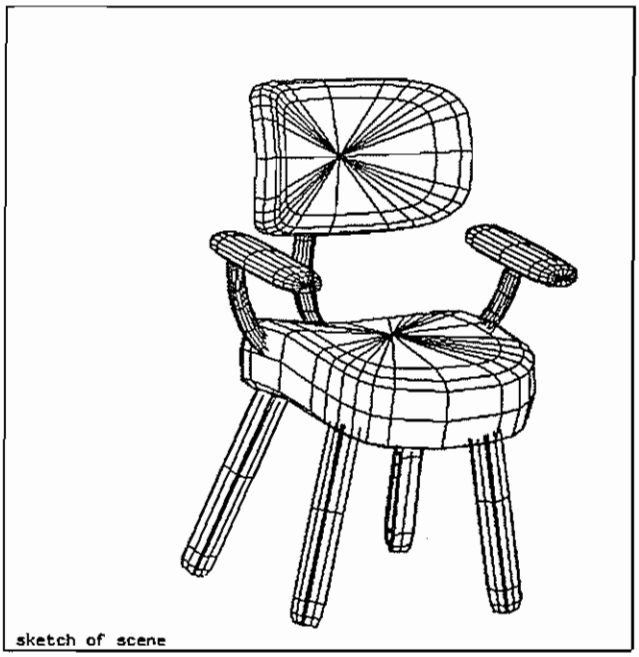
Following in this tradition, we have examined [13] the manner in which people describe shape by using both the tools of protocol analysis and the psychophysical method devised by Triesman. One concise characterization of our results is that we observed our subjects describing 3-D shapes *procedurally* — namely, by describing how one would make the shape using a malleable material such as clay, using a few generic forming actions [14]. As an example, our subjects might have described the back of a chair as a rounded, flattened



A



B



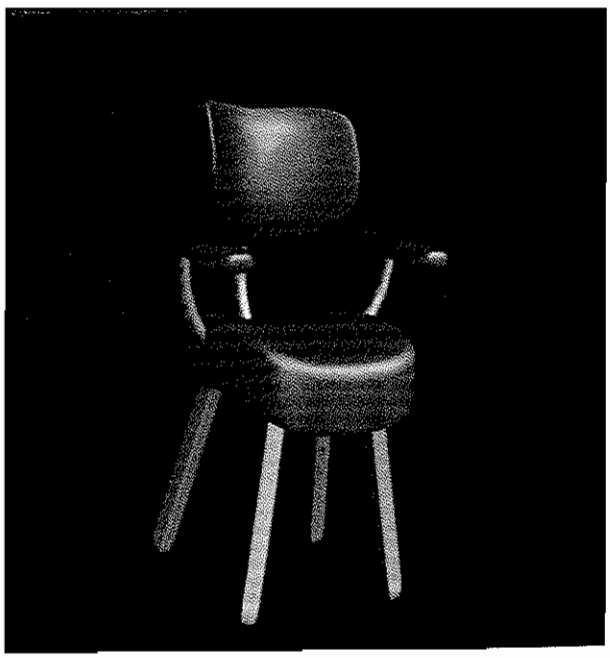


Figure 2: (a) A sampling of the basic forms allowed, (b) deformations of these forms, (c) a chair formed from Boolean combinations of appropriately deformed superquadrics.

cube, that has been slightly cupped or bent to accommodate the human form. The bottom of the chair might be described as a similar object, but rotated 90°. By "oring" these two parts together with elongated rectangular primitives describing the chair legs, they would obtain a complete description of the chair. This description is illustrated in Figure 2(c).

One reason people may favor such a part-and-process method of description is that it offers considerable potential for reasoning tasks. It seems, for instance, that people employ such descriptions in learning as well as in commonsense and analogical reasoning [15-17]. This may be because such descriptions refer to the world in something like "natural kind" terms: they speak qualitatively of whole forms and of relations among the parts of objects, rather than of local surface patches or of particular instances of objects.

Moreover, recent research in graphics, biology, and physics has given us good reason to believe that it may be possible to *objectively* describe our world by means of a few, commonly occurring types of formative processes [18-21]. In essence, we think that our world can be modeled as a relatively small set of generic processes — such as bending, twisting, and interpenetration — that occur again and again, with the apparent complexity of our environment being produced from this limited vocabulary by compounding these basic forms in myriad different combinations.

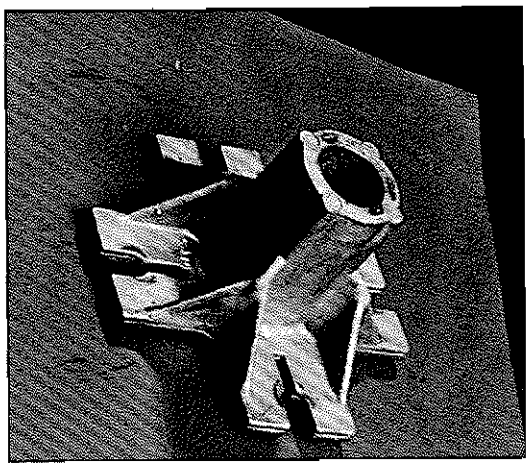
2.1 Our Representation of Part Structure

Inspired by the way people seem to describe shape, we have adopted a representation that describes shape in a similar manner — that is, by explaining how one would create a particular shape by forming and combining lumps of clay. The most primitive notion in this representation is analogous to a "lump of clay," a modeling primitive that may be shaped and deformed, but which is intended to correspond roughly to our naive perceptual notion of "a part." For this basic modeling element we use a parameterized family of shapes known as a *superquadrics*, invented by Danish designer Peit Hein [22,23]. These are described (adopting the notation $C_\eta = \cos\eta$, $S_\omega = \sin\omega$) by the following equation:

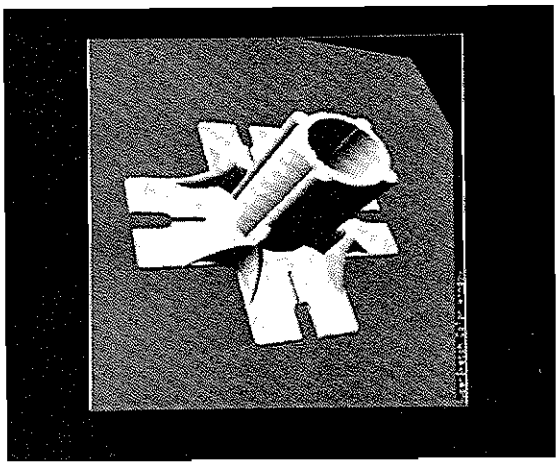
$$\vec{X}(\eta,\omega) = \begin{pmatrix} C_\eta^{\epsilon_1} C_\omega^{\epsilon_2} \\ C_\eta^{\epsilon_1} S_\omega^{\epsilon_2} \\ S_\eta^{\epsilon_1} \end{pmatrix} \tag{1}$$

where $\vec{X}(\eta,\omega)$ is a three-dimensional vector that sweeps out a surface parameterized in latitude η and longitude ω, with the surface's shape controlled by the parameters ϵ_1 and ϵ_2. This family of functions includes cubes, cylinders, spheres, diamonds and pyramidal shapes as well as the round-edged shapes intermediate between these standard shapes. Some of these shapes are illustrated in Figure 2(a). Superquadrics are, therefore, a superset of the CSG modeling primitives that are currently in common use.

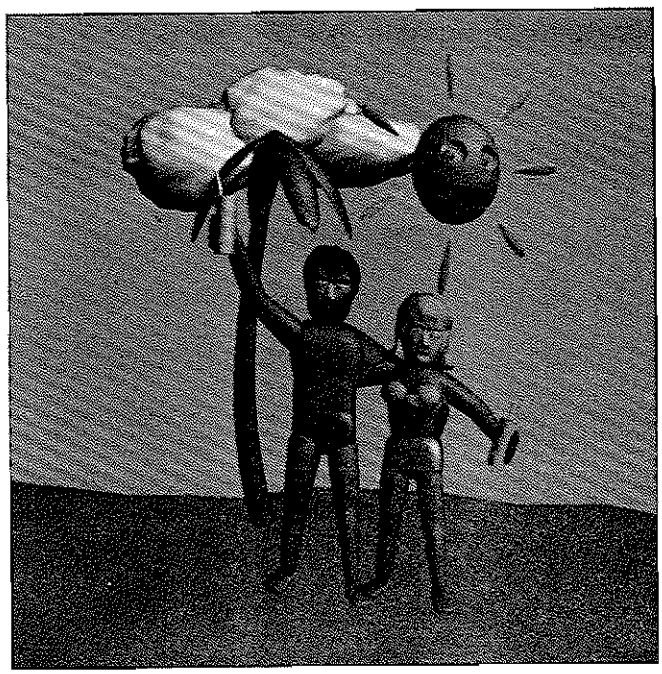
These basic "lumps of clay" are used as prototypes that are then deformed by linear stretching and tapering, or by quadratic bending, and finally combined through Boolean operations to form new, complex shapes. We have found that this representational system has a surprising generative power that makes it possible to create a wide variety of form, as illustrated in Figures 2 and 3. Interestingly enough, this representation turns out to be both a generalization of the constructive-solid-geometry approach and a modification of the



A



B



C

Figure 3: (a) An industrial casting, (b) Its SuperSketch model (approximately 300 parameters; construction time: 23 minutes), (c) A SuperSketch model of two people (approximately 1000 parameters; construction time: about 4 hours)

generalized cylinders approach; we are combining a restricted class of generalized cylinders through Boolean operations.

We have constructed a 3-D modeling system, called "SuperSketch," that employs this shape representation. This real-time, interactive modeling system, implemented on the Symbolics 3600, allows users to create "lumps" interactively, to change their squareness/roundness, to stretch, bend, and taper them, and finally to combine them through Boolean operations. We have found that, with SuperSketch, users can quickly model a wide range of man-made and natural shapes in a way that correctly captures the intuitive part structure of the object. This system was used to make the images in this paper.

2.2 Implications for Machine Vision

Perhaps the most important fact about this representation is that it uses a small number of parameters to describe a surprisingly general-purpose class of 3-D modeling primitives, thus enabling us to describe hundreds of image pixels with only 14 parameters.[1] This compares favorably with the nine parameters needed to describe intensity, first, and second derivatives at a single point; the nine parameters needed mearly to describe the position, orientation, and size of a rectangular solid; or the hundreds of parameters that might be needed to describe an equally detailed generalized cylinder. As a consequence of the concise nature of this shape language, we have found that even complex objects and scenes can usually be modeled by using a relatively small number of parameters; for example, the models shown in Figure 3 have only a few hundred parameters each.

These modeling results have led us to believe that this representation provides a concise "natural" level of description — exactly the sort needed to support recognition. The fact that this representation can describe a wide range of 3-D shapes by using a relatively small number of parameters leads us to believe that it may provide the constraint we need to be able to search for optimal descriptions of shape without a fatal combinatorial explosion.

Given such a "part" representation, we must now develop techniques for recovering these "parts" from image data. There are two cases to consider: (1) the directed, top-down case in which we expect a certain object to occur near some particular location; (2) the bottom-up case in which we must compute a new description solely on the basis of raw image data. The following discussion will be with reference to range data only, as the primary sensor now being used by the ALV is a laser rangefinder.

3 LEARNING NEW OBJECT DESCRIPTIONS

Reliable, bottom-up learning of object descriptions is perhaps the most difficult task faced by a vision system. Certain characteristics of range data, however, make the problem much simpler. The primary simplifying characteristic is the fact that range imagery allows simple separation of figure from ground: one "chops out" a cube of data, locates the ground plane, and the remaining data is "figure." The technique we are currently using to separate figures

[1] Three for position, three for orientation, three for size, two for squareness/roundness along the various axes, two for bending and one for tapering. We restrict bending to being along at most two axes, one of which is the longest axis, and tapering to only the longest axis. These restrictions stem from our finding that additional degrees of freedom were almost never used when people constructed models with SuperSketch.

from ground is described in [25]; briefly, is a matter of fitting a "rubber sheet" model to the surface observed in the range data and then finding those places where the surface changes elevation rapidly enough to "tear" a hole in the sheet.

Having identified patches of image data as obstacles that need to be described and having adopted a particular representation, we can now pose the problem of learning an object description as the process of using this shape vocabulary to find the "best" account of the data. That is, we can define the learning process as one of optimizing our description over our shape vocabulary relative to some goodness-of-fit criterion. We shall employ use minimal-length in this paper as our means of evaluating an explanation of the image data, in other words, our goodness-of-fit criterion is a linear combination of the data error (evaluated by using the L_1-norm) and the number of parameters in the description.

Minimal-length encoding is a natural way to capture the intrinsic structure of a scene by examining the image data. One can, for instance, prove that if a body of data is generated by a shape vocabulary V with parameter settings P_i, then the minimal-length encoding of that data (using V) will recover the P_i — given sufficient resolution, noise-free data, and modulo ambiguities in the vocabulary. [2]

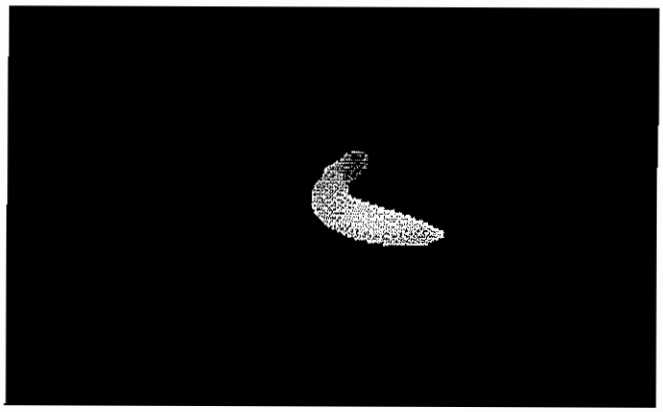
Thus, the important point about employing this approach to learn a description is that we must use a vocabulary that correctly describes the actual structure of the data. Our choice of vocabulary is based upon psychological evidence as to people's perception of the intrinsic structure of 3-D objects [10-18].

The primary difficulty in computing a minimal-length encoding is that it requires global optimization of a fitting function and, unfortunately, there are no efficient, general-purpose global optimization techniques for such nonlinear problems. We may, however, take advantage of the special properties of this particular problem in order to achieve an adequate solution.

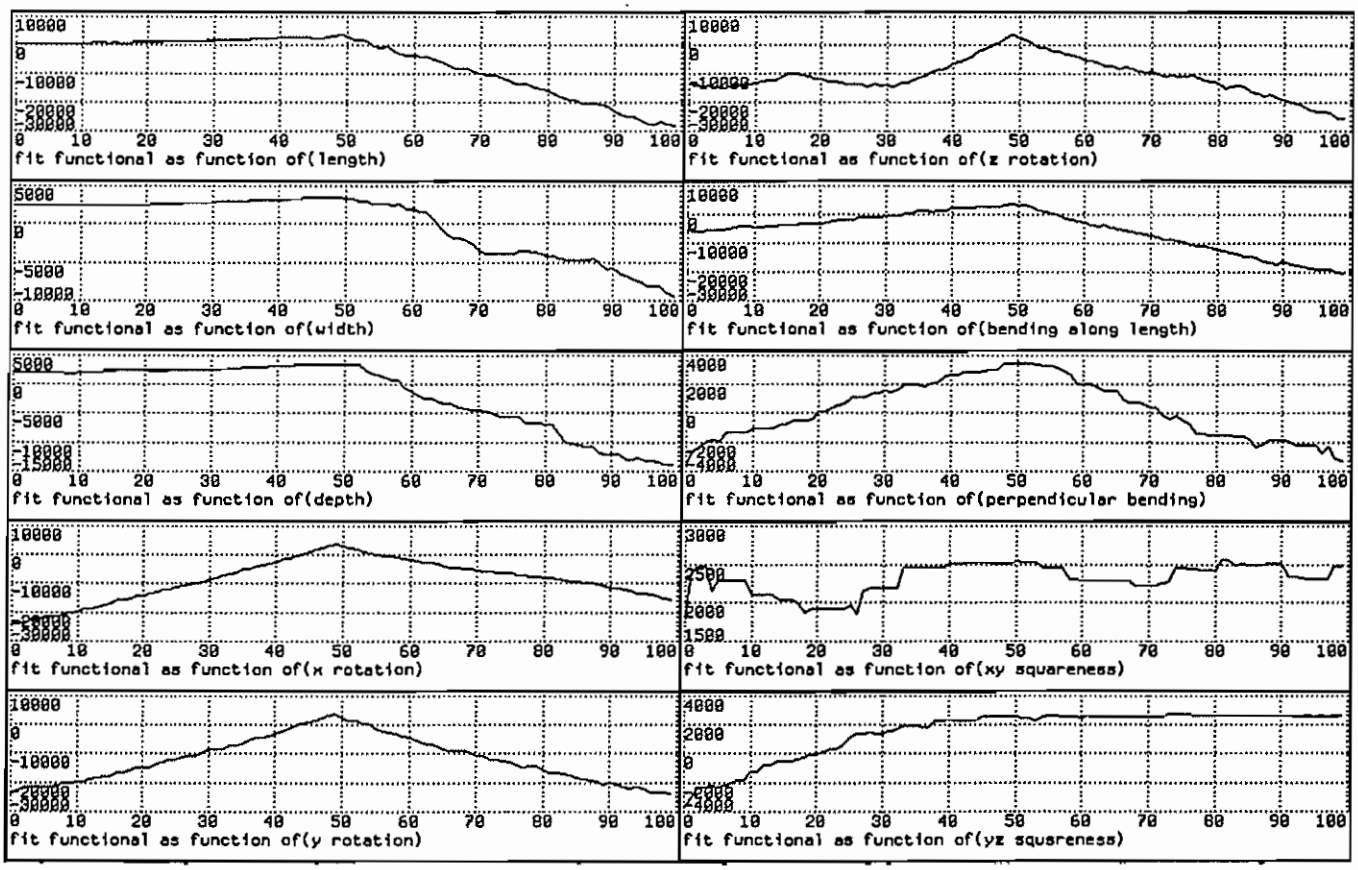
The first property we may take advantage of is that we may decompose of our search for the best explaination into two phases: a local phase and a subsequent global phase. We may do this because the part models in our shape vocabulary are compact, with surfaces that are opaque to the sensor. Thus, changes that are far enough from a particular image point do not affect the description at that point. For instance, if the largest element in our shape vocabulary has a projected radius of 50 pixels, we do not have to look more than 60 or 70 pixels in any direction in order to find the part model that provides the best fit in the immediately surrounding image region.

The second property we may take advantage of is that the search for a "best fit" within a particular image region seems to be convex for parameter values near the optimal solution. Figure 4(a) shows a range image of a bananalike shape. Figure 4(b) shows the value of an L_1-norm "goodness-of-fit" functional between these range data and a 3-D SuperSketch "part" as each of the part's parameters are varied. At the center of each graph in Figure 4(b) is the exactly matching parameter value; it can be seen that our "goodness of fit"

[2]Briefly stated, if we have no ambiguities in the vocabulary — e.g., if we disallow situations in which one vocabulary primitive can be fitted perfectly by some combination of other vocabulary primitives — we can express the image data as an overconstrained set of equations whose variables are the generating primitive's parameters. If we change any of the parameters, the encoding is no longer accurate and therefore not minimal; if we add a primitive, the encoding is no longer minimal; consequently, because the system has no ambiguities and is overconstrained, there is no accurate encoding containing fewer primitives.



A



B

Figure 4: The fitting problem seems to be locally convex. (a) A range image of a banana-like shape, (b) The fit between these range data and a 3-D part model as the parameters of the 3-D part are varied; the correct fit occurs at the center of each graph.

functional varies slowly as we move away from the correct parameter value (note that these graphs do not show the functional's value over the entire parameter range).

Figure 4, therefore, illustrates our empirical observation that finding the "best fit" within a neighborhood is a convex problem for parameter values near the optimal solution. In our experience this convex behavior obtains uniformly. The convex region surrounding the correct parameter setting is often quite broad: For instance, we can usually vary length, width, or depth by up to 50% before we leave the convex region of the parameter space. In contrast, the "goodness of fit" functional is relatively sensitive to orientation: We normally can vary rotation angles by only 10 or 15 degrees before leaving the convex region.

These two properties, together with small number of parameters in our modeling primitives, means that we can use a coarse-grained search over the entire parameter space as our optimization procedure. That is, by comparing (using an L_1-norm) each combination of parameter settings with the image data surrounding each image point, we can find a small set of part models which provide the best *regional* fit to the image data. We can then search among combinations of these regional best-fits to find the best overall explanation of the image data, i.e., a minimal-length encoding of the image data in terms of our modeling primitives.

We have found that, if the (x, y, z) position parameters are excluded, about 84,000 "goodness-of-fit" evaluations are required to search the entire parameter space adequately, sampling most parameters at three different values (e.g., object widths of 10, 20, and 40 inches) and some critical parameters, such as orientation, more frequently (e.g., every 22.5 degrees) [3]. The experimental finding that our goodness-of-fit functional varies slowly near the correct parameter setting gives us confidence that we will find a parameter setting that is within the convex region surrounding the optimum parameter setting. We may then refine the result of this coarse search by using a gradient descent algorithm.

By repeating this search for each [coarsely quantized] image position, *we can effectively carry out a coarse-grained search over the entire parameter space.* This produces a relatively small set of part models that provide the best *regional* fit to the image data. We then choose from among this set of regional fits a small set of parts that provides the best explanation (i.e., a minimal-length encoding) of the image data, and, finally, conduct a gradient descent on this final ensemble of part models.

3.1 Evaluation of the Goodness-Of-Fit Functional

One of the key elements assuring the success of this approach is evaluation the goodness-of-fit functional in a manner that is insensitive to occlusions and that, in addition, takes into account all the information we have about edge placement, surface shape, perspective effects, and sensor characteristics. The procedure we use is to

(1) Construct a 3-D SuperSketch part with the hypothesized parameters (e.g., orientation, length, squareness).

(2) Render that part by using known sensor characteristics, thereby constructing, e.g., a range image, that accounts correctly for the effects of perspective, surface shape, and

[3] In our sampling we restrict bending to occur along at most two axes, one of which must be the longest axis, and we have typically not included tapering (it seems to make a difference only on large forms), except when using a reduced sampling in orientation.

other known variables that affect image appearance.

(3) Histogram all of the point-by-point differences in depth between the rendered part and the image data. This produces a histogram with "buckets" at each possible offset between the hypothesized part's depth and the actual depth, so that the value in each bucket is the number of pixels at that particular offset. Note that we use the L_1-norm in this operation. While doing this, we also keep track of the number of pixels ϵ that fall off the figure entirely (when the "figure" of interest can be separated from the surrounding "ground," as is generally possible with range data).

(4) From this histogram we estimate the position p of the largest peak. This peak is the most frequent distance between the hypothesized and the measured surfaces; the number of counts in this peak is the area (in pixels) of the hypothesized part's surface that would match the image data if the part were moved in depth by a distance p. By using buckets of width σ, therefore, we can employ this histogramming technique to determine the number of pixels μ that would match to within $\pm\sigma$ at the optimum depth positioning of the hypothesized part.

(5) Compute the value of the goodness-of-fit functional Γ for this set of parameters:

$$\Gamma = \mu - \lambda\epsilon \tag{2}$$

The quantity $1/2\Gamma\sigma$ is our L_1-norm estimate of the total fitting error between the hypothesized part model and the observed data over the region in which the L_1-error is small enough to make a match plausible. In the following examples, $\sigma = \lambda = 4$; the procedure seems to be relatively insensitive to the value of these parameters.

By using a rendering technique that includes a full camera and sensor model, we make *explicit* all of the edge, surface, perspective, and other relations that are normally *implicit* in our model parameters. We are thus able to take into account these previously implicit relations.

By using a RANSAC-style [24] histogramming procedure, we allow large portions of the figure to be occluded without disturbing the matching process. This enables us to compute an energy functional that corresponds closely to the L_1-norm error between the image data and the hypothesized shape, while ignoring data that are due to other, occluding forms.

3.2 Refining the Initial Search

Within each region searched, the above procedure gives a "best-fitting" 3-D SuperSketch part model for the image data in the surrounding region. This coarse-grained global search is then followed by a gradient-descent optimization of the part's parameters versus the goodness-of-fit functional. We use a stochastic technique to avoid shallow local minima; our method is similar to the one employed by Bajcsy and Solina [31].

A single "best" explanation for the entire body of image data is then found by picking a minimal covering of the data from among the set of regionally best-fitting part models. Currently this is accomplished through a simple iterative, best-first search: We first accept the 3-D SuperSketch part model that accounts for the most image data, then the part that accounts for the maximum portion of the remaining data, and so forth. When we accept a part, therefore, those parts centered at nearby locations are generally excluded from further consideration because they are "covered" by the already accepted part. This search

technique was chosen because, although its performance is almost always suboptimal, it is efficient and its typical-case performance is good.

This procedure, therefore, gives us a set of part models that furnishes a reasonable approximation to the minimal-length encoding of the image data in terms of our part representation. We have found, however, that it is useful to perform a final stochastic optimization on all of the parameters of this final set of parts because the histogramming technique described above is not completely insensitive to occlusion relations. We accomplish this final optimization by means of a numerical gradient descent algorithm that renders all the hypothesized part models together (thus completely accounting for occlusion relations) and computes the goodness-of-fit functional; the algorithm then changes one of the part's parameters and ascertains whether the fit is improved. If improvement does indeed occur, the change is accepted..

3.3 Practical Considerations

Straightforward implementation of the above operations requires about 10^9 operations per image region. The number of operations required can be reduced considerably by pruning the search space during the search, in a manner similar to that used by Bolles [5], Goad [7], or Grimson and Lozano-Perez [9] in their model-based, global-search-and-match vision systems.

In our current implementation, this pruning is accomplished by keeping track of the current n best fits (largest Γ values) within an image region and by using their Γ values to (1) abort evaluations of Γ as soon as it becomes clear that the eventual value will be smaller than any of the current n largest Γ values (e.g., when, even if all of the remaining pixels match exactly, Γ will still be smaller than the current n best Γ values), (2) discard any parameter setting that *a priori* cannot generate a value of Γ that is larger than one of the current n best Γ values, and (3) order the search so that the above pruning techniques will be maximally effective (e.g., search over the parameter settings with the largest potential Γ values before searching over other parameter settings).

By taking advantage of these and other efficiency expedients, the examples shown here have required an average of about 10^{10} operations each, roughly two and one-half hours of CPU time on a Symbolics 3600. For industrial applications, in which bending and tapering are not typical, the search space is smaller and therefore the required computation time can be as much as $1/30^{th}$ that of the full algorithm. Because of the inherent parallelism of the technique (thousands of identical searches within each region) a full global search is expected to take approximately one seconds per image with today's large, parallel computers.

Perhaps more importantly, however, a fully developed system would also make use of such image features as edges and corners to prune the search space. Additionally, hierarchical coarse-to-fine techniques could be used to guide the search, thus improving efficiency and perhaps eliminating the need for gradient-descent improvement at the end of the global parameter-space search. We have not as yet had time to explore these possibilities.

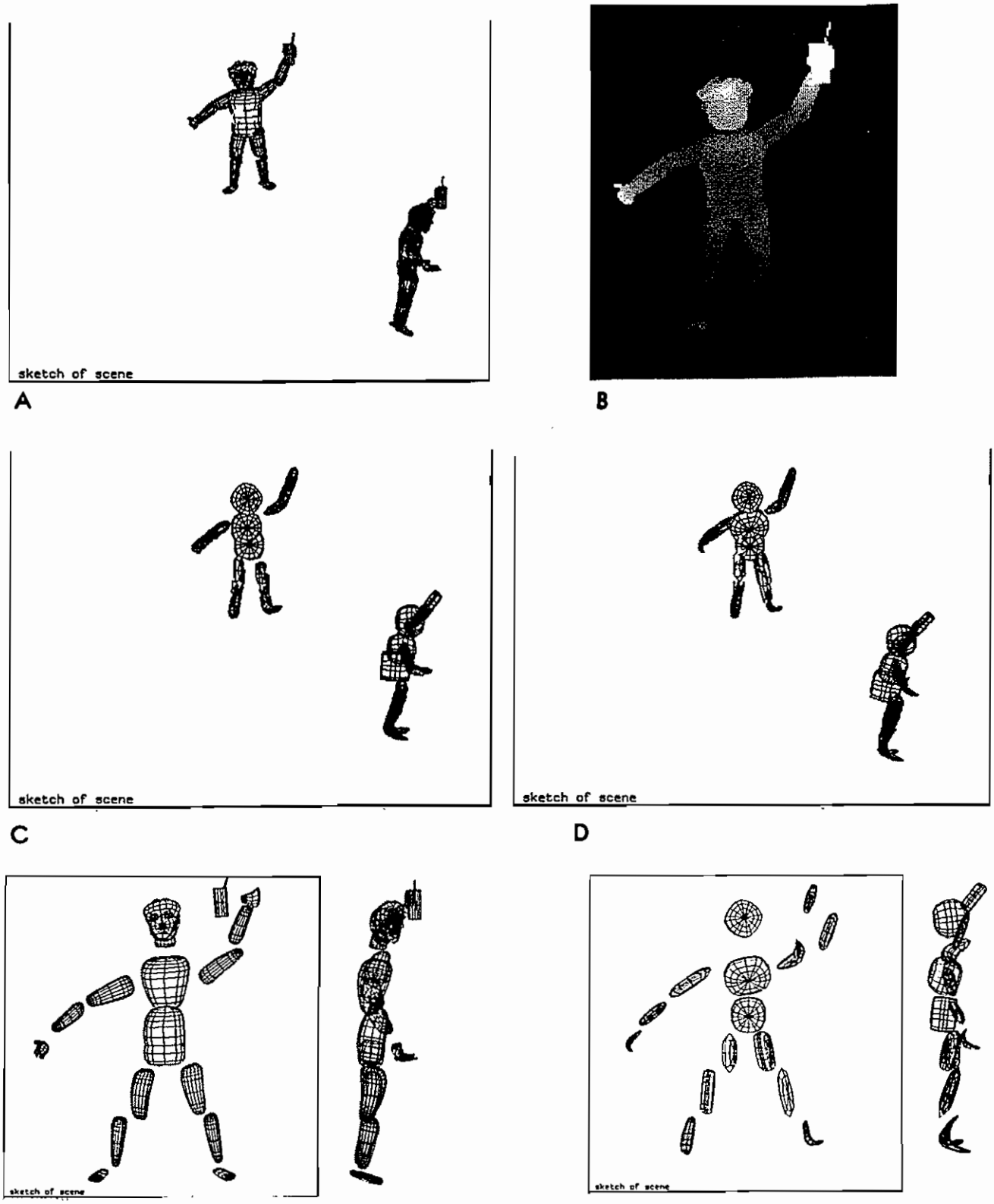


Figure 5: (a) A SuperSketch model, (b) a range image generated from this model, (c) Initial explanation of the image data, (d) Final learned model, (e) "blow-up" views of the original model and the learned model. Note the similarity of part structure and part-by-part parameters.

4 LEARNING RESULTS

One of the major advantages of the above global search technique is the certainty of convergence to a good answer (in terms of accounting for the metric properties of the data) within a fixed period of time. It may be more important, however, that we obtain an stable account of the object's *part structure*, for this is what we will use to determine the object's class identity (e.g., "gate," "car") for purposes of reasoning. Whether a model that accounts accurately for metric properties also accounts for part structure depends upon whether our shape vocabulary actually expresses a robust aspect of the object's true 3-D structure, as well as whether that structure is conserved under projection. This issue is one we will address in this section.

The following examples involve range imagery; for purposes of evaluation we present one synthetic example, followed by three examples using the ALV's laser range-finder. One characteristic of range data is that is it generally easy to obtain a rough "figure/ground" separation [25]; we have taken advantage of that ability. Some simple preprocessing of the laser rangefinder data was used to remove mixed-range pixels as well as the inherent ambiguity-interval problems of those data [25,26].

4.1 A Synthetic Data Example

The first example uses simulated range data, with approximately six-bit resolution. The purpose of this example is to demonstrate (1) the performance of the algorithm independent of special data characteristics, and (2) the ability of the technique to recover part structure despite the problems of scale and configuration.

Figure 5(a) shows a SuperSketch model (here and in succeeding figures we will show side views of SuperSketch models as insets placed in the lower-right-hand corner of the frame surrounding the model); Figure 5(b) shows a range image generated from this model. This is a fairly accurate model of the articulated human form; as such, it illustrates the necessity for a part-structure representation of the overall shape: without such a representation, we would have to store descriptions of every possible positioning of the figure in order to recognize it as it moves about. Figure 5(c) shows the initial explanation of the image data, found by iterative, best-first search among the best fits at points sampled along the figure's 2-D skeleton.

The main thing to note about this initial shape description is that, although not perfect, it seems sufficiently similar to the original, generating description that we can use it to index into a database of known forms and to recognize the figure as human.

Figure 5(d) depicts the final learned model, the result of gradient descent from Figure 5(c). Figure 5(e) shows "blow-up" views of both the original and the learned model. Note the similarity of part structure and of part-by-part parameters.

Perhaps the major question to be asked about this procedure for recovering part structure is whether or not the recovery is *stable*, since stability in structuring the image data – i.e., in producing a *segmentation* — is the primary property required for reliable higher-level reasoning functions such as class formation and generalization [16,35,36]. Some of the particulars of this example are illustrative in this regard.

Note, for instance, that the feet are described by bent primitives that account for both ankle and foot, even though in the final model the "ankle" part is not visible, having been

occluded by the "calf." Such fitting of a bent primitive to two unbent parts is also observed in the right arm — here too, the upper part is occluded, by the "forearm." Although these assignments of part structure are perhaps, not perfect, they are entirely plausible segmentations when only one view is given. Such occasional merging of connected primitives seems unavoidable with only one view; thus, we must allow for *both* possible descriptions — as one bent primitive and as two straight but connected primitives — when forming object classes or searching for similar stored models.

A more interesting case occurs in the recovery of descriptions for the hands and head, for although both head and hands are actually quite complex shapes, they are recovered as being a single, undifferentiated part. These examples show the effect of *scale*; when the image features become smaller than the range of scales searched, there is a sort of "summarizing" effect as a fit is attempted to the overall composite form. Marr and Nishihara pointed out the need for this type of "summarizing"; they proposed that we must employ a *multiscale* representation in comparing the learned model with stored models. In this example we can see how a multiscale representation, with descriptions for each distinct scale of part structure, might be combined with this recovery procedure to resolve some of the difficult problems associated with scale.

4.2 Learning Descriptions Outdoors

The remaining examples make use of data from the ALV's ERIM time-of-flight laser range finder. This rangefinder, which collects a 256 x 64 pixel image in 0.4 seconds, has a useful range of about 128 feet and an advertised accuracy of about five percent. Its unusual imaging geometry is similar to that of a very-wide-angle lens.

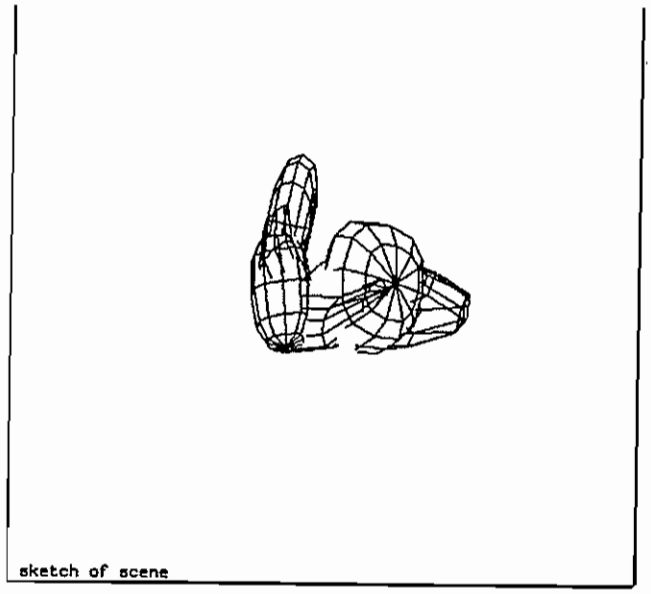
Figure 6(a) shows a range image of the upper part of a famous industrial vision researcher, taken with this sensor. This example is interesting, especially in comparison with the synthetic data example above, in that the amount of depth information within the figure is negligiable; from a practical point of view, this is merely a silhouette. Figure 6(b) shows the initial explanation of the image data, while Figure 6(c) displays the result of gradient descent from Figure 6(b).

Perhaps the most salient point of this example is that a reasonable 3-D part structure can be learned even from what is essentially only silhouette data; the left (upraised) arm, head, and right arm are clearly present in the learned description. Figure 6(d) shows a comparison of the original range data with a range image produced by the learned model of Figure 6(c). This comparison shows that the simple (70 parameter) learned description retains most of the data's original metric information, once again demonstrating descriptive adequacy.

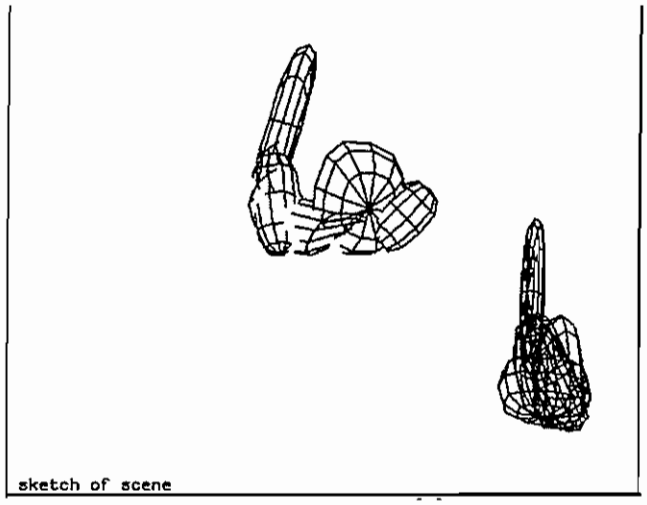
Figure 7(a) shows a range image of a gate by the side of the road. Again, the data contain little more information than a silhouette; the linear elements of this figure average two pixels across. Note that our figure/ground procedure has inadvertently included a bush near the left-most gatepost as part of the figure. The most interesting aspect of this example is the small size of the imaged features; these data thus provide a severe test of noise sensitivity.

Figure 7(b) shows the initial explanation of the image data. Once again a reasonable part-structure description is learned, although the left gatepost is seen as a block reflecting

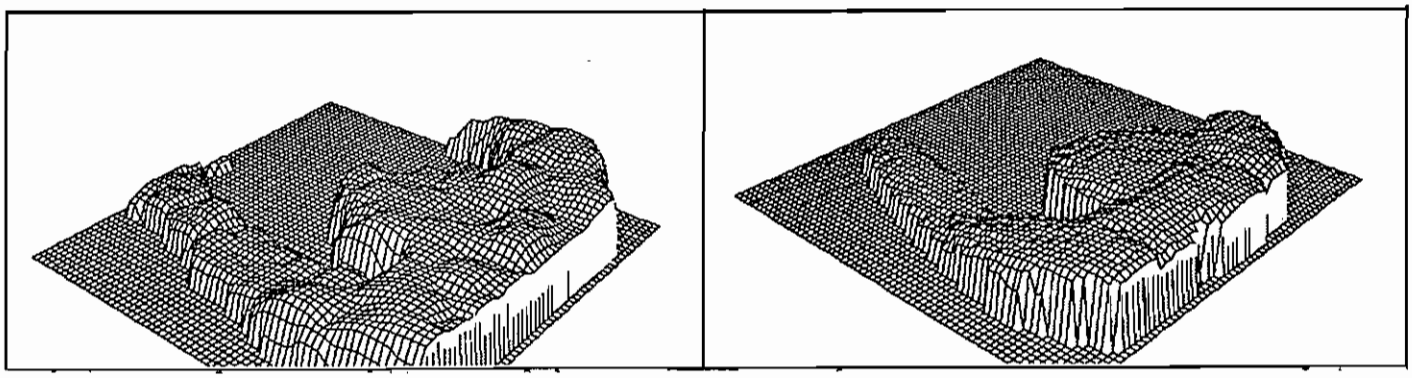
A



B



C



D

Figure 6: (a) A range image of the upper part of a famous industrial vision researcher (for practical purposes this is merely a silhouette); (b) the initial explanation of the image data (note that the correct part structure of his left (upraised) arm, head, and right arm are clearly present); (c) the final learned model; (d) comparison of the original range data with a range image produced by the final learned model.

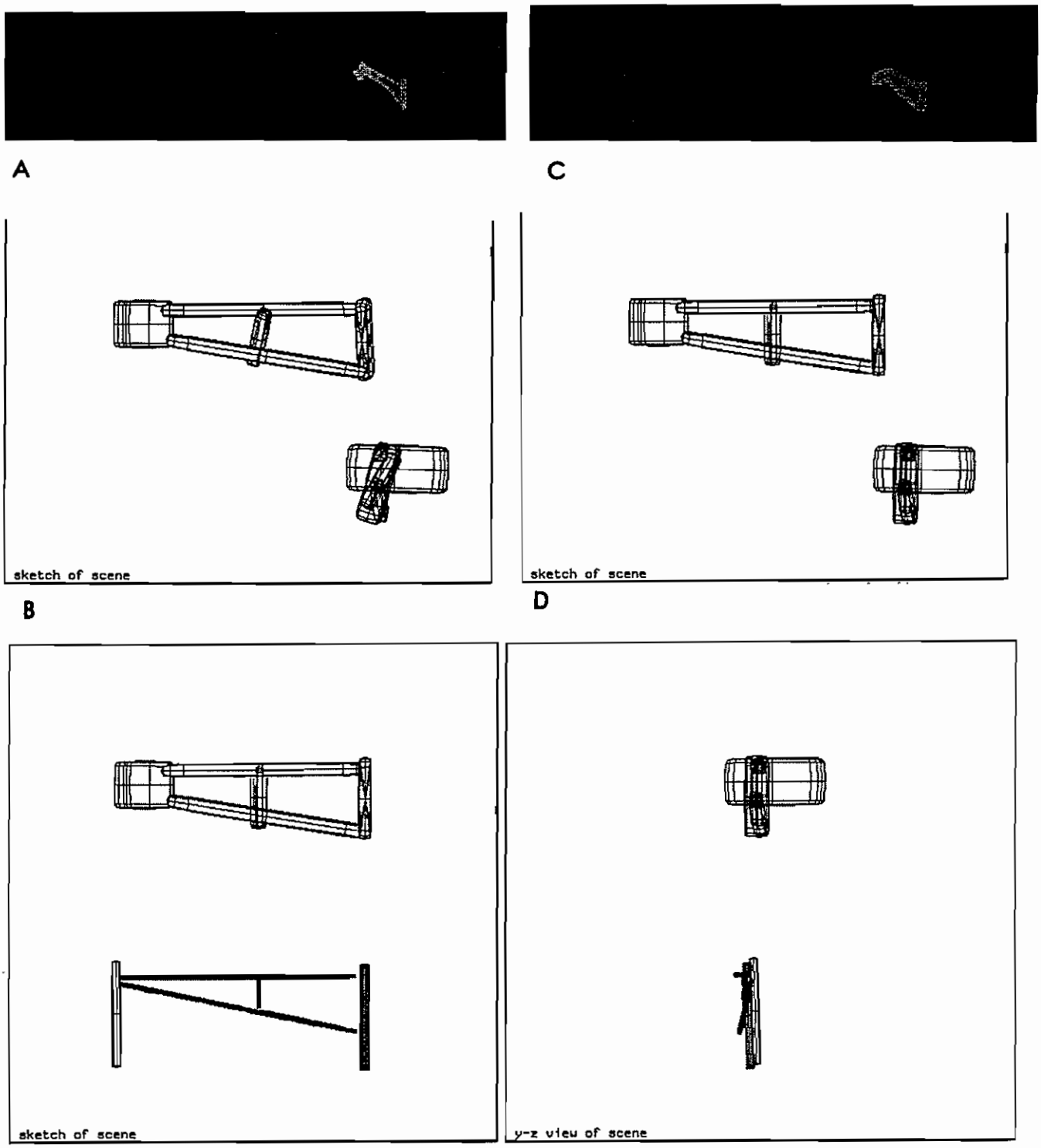


Figure 7: (a) A range image of a gate by the side of the road (our figure/ground procedure has unintentionally included a bush near the leftmost gatepost as part of the figure); (b) the final learned model; (c) a range image produced by the learned model; (d) The result of "prettifying" (b) using the domain knowledge to the effect that nearly horizontal/vertical parts are likely to be horizontal/vertical; (e) comparison of the learned model of (d) with a SuperSketch model of the gate that was constructed by hand.

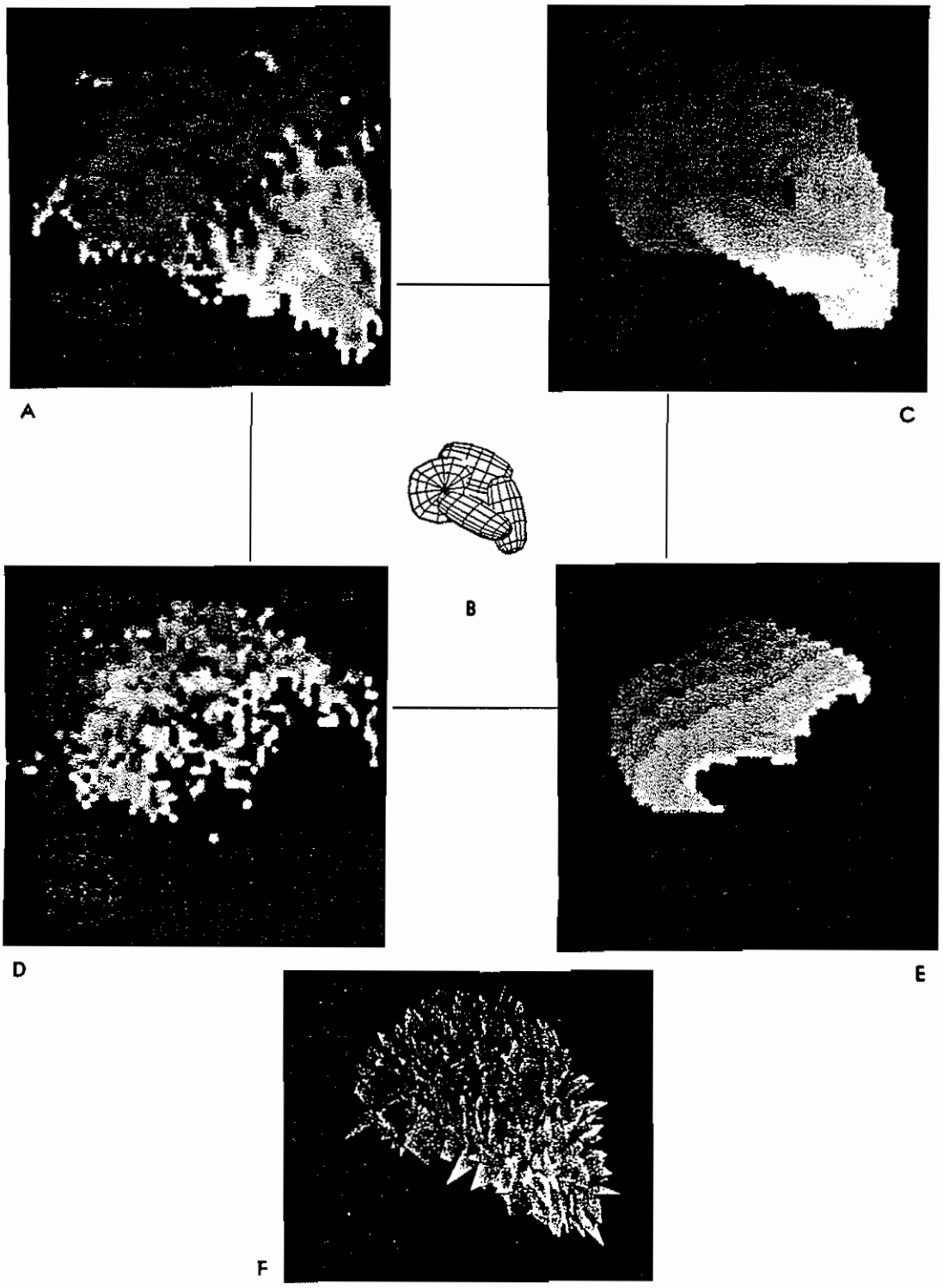


Figure 8: (a) A range image of a few roadside bushes; (b) final learned model; (c) A range image produced by the learned model; (d) the original range data, with points closer than the median distance removed; (e) A range image produced by the learned model again with points closer than the median distance removed, showing the model's close match with the range data's internal structure; (f) range image produced by adding a fractal surface model to the part representation.

the fact that the data include a bush as well as the gatepost. Figure 7(c) contains a comparison of the original range data with a range image produced by the learned model of Figure 7(b); here too, the learned description retains most of the data's original metric information.

One of the advantages of having a high-level description like this part language is that it provides good "hooks" for applying domain-specific knowledge. This is illustrated by Figure 7(d), which shows the result of "prettifying" Figure 7(b) by making use of the domain knowledge that nearly horizontal or vertical parts are likely to be horizontal or vertical. Figure 7(e) contains a comparison of the learned model of Figure 7(d) with a manually constructed SuperSketch model of the gate. The thickening of the posts and bars in the learned gate can be largely attributed to preprocessing intended to remove mixed-range pixels.

Figure 8(a) shows a range image of a few roadside bushes. The most important aspect of this example is that it is *not* an example in which there is an obvious part structure; nor is it an example with smooth surfaces. These data, therefore, allow us to examine some of the limits of our technique's descriptive adequacy, as well as its ability to produce stable segmentations. Figure 8(b) shows the initial explanation of the image data, as found by iterative, best-first search among the best fits at points in a 10 x 10 grid covering the image data.

Figure 8(c) shows a range image produced by the learned model of Figure 8(b); the outlines of Figures 8(a) and (c) can be seen to be similar, thus confirming that the learned description (in this example a total of 56 parameters) retains most of the data's original metric information. Figure 8(d) shows a comparison of the original range data, with points closer than the median distance removed and a range image produced by the learned model of Figure 8(b), again with points closer than the median distance removed. This comparison shows that the *internal* structure of the learned model closely matches that of the measured range data.

This example shows that even very complex shapes can be usefully "summarized" by our shape vocabulary; thus supporting our dual claims of descriptive adequacy and stable part structure recovery. We can improve the accuracy of the learned model somewhat by modeling the discrepancies between the recovered part structure and the range data by using a fractal surface model [28,32]. The result of adding a fractal surface model to the recovered part model is shown in Figure 8(e). An even better description of the differences between part model and image data could be obtained by using a particle model [33] of the bush's branches and leaves.

4.3 Summary

We have described a representation that is fairly general purpose, despite having only a few parameters[4]. Capturing this expressive power with a small number of parameters has allowed us to approach the problem of learning object descriptions using the model-based

[4]We note that *any* part structure representation must have a *least* nine parameters: three each for position, orientation, and size. Our representation adds only five more parameters in order to achieve its reasonably general-purpose descriptive power. We note that the fact that we could use our modeling vocabulary to obtain accurate fits to the data in these examples lends support for its descriptive adequacy.

vision technique of global search-and-match. On the basis of our experiments we believe that our method is quite robust; our examples, for instance, confirm that the recovery procedure is reasonably stable with respect to noise and scale.

This system does not now make use of feature-level cues to either determine part structure or estimate parameter values. Nor does it make use of hierarchical, coarse-to-fine strategies to guide the search. It seems clear that adding such capabilities would only enhance to the efficiency and robustness of the algorithm.

One reason for restricting our attention to image-level matching is to produce an algorithm that can be applied directly to the outputs of shape-from-x and depth-from-x vision modules. We believe that our current algorithm would be effective in such an application.

5 RECOGNIZING OBJECTS

The previous section showed that, by using a part representation, one could "bottom-up" compute an accurate description of interesting objects from ALV range imagery. Having accomplished the task of learning the position and description of various landmarks, our vehicle is now faced with the very different problem of using these learned descriptions to navigate.

In the navigational problem the flow of information is largely top-down, in contrast to the learning task in which the information flow was almost exclusively bottom-up. This is because in the navigational task, we already know the landmark's approximate position, and have a description of it in our database. Thus, we can simply "monitor" the appearance of the landmark — that is, we can update our knowledge of vehicle position and landmark structure through a simple gradient descent procedure that compares predicted appearance with measured appearance, adjusting position and structure to minimize any discrepancy.

The main requirements for our recognition procedure, therefore, are that it be able to (1) recognize the landmark from any vehicle viewpoint, (2) handle small errors in predicted orientation (e.g., ± 20 degrees), predicted position (e.g., ± 15 feet), and our object's description (e.g., ± 2 feet in height or relative position). If our recognition algorithm fails, it should be capable of notifying us, so that we can invoke the more expensive bottom-up learning procedure, or, alternatively, a hybrid procedure that searches for parts that are roughly similar to those that were expected.

Such a monitoring function has the advantage that is can be quite efficient: the algorithm described here, for instance, could be executed in a few dozen seconds on a Symbolics 3600. Thus, when the vehicle is traveling in previously traversed areas, most of the its computational power would be available for other tasks. It is only in unknown terrain that significant computation is required.

5.1 Recognition by Image-Level Predicition

The technique we are using at present for landmark recognition is based on predicting the appearance of each of the landmark's component parts in the range data. In most machine vision applications data-level matching is risky because it is difficult to predict sensor values. However, since this sensor produces range (instead of intensity), it is plausible to match directly to the image values. Additionally, as before, we hope that, by producing

an algorithm that matches directly to range data, we will obtain a recognition technique that can also be applied directly to the outputs of shape-from-x and depth-from-x vision modules.

We are now capable of taking the following variables into account when making our predictions:

- The range sensor's spherical geometry.
- The motion of sensor (as measured by an on-board internal navigation system).
- "Mixed pixels" along object boundaries, which we account for by predicting image appearance at twice the normal resolution and then subsampling appropriately.
- Sensor ringing and blur, again accounted for by appropriate subsampling of a super-resolution predicted image.

By taking these effects into account we have found that we can predict sensor range values with sufficient accuracy.

Using this sensor model, we construct two "masks" that give (1) the expected pixel-by-pixel range values for each *part* in our model of the object and (2) the pixel-by-pixel expected variance for the predicted range values. We then perform individual searches for each *part* in our model of the object, using the first of these masks to compute the sum-of-squares difference between the expected range values and the actual range values, with the second mask serving as a set of weights for combining these differences to form a weighted sum-of-squares fit for each part at every point in the image. We then factor in our knowledge of the vehicle's position and the object's interpart geometry to obtain least mean square error estimates for the object's location and geometric structure.

By modeling the error in position, sensor sampling, quantization, and noise, we are able to account for the primary known sources of error in our model of the landmark's appearance. Perhaps even more important, however, is the fact that by performing the search on a part-by-part basis enables us to minimize the effects of unknown *global* distortions, such as those that arise from small deviations in viewing angle or from errors in the model's geometry. This is because parts are smaller and have less structure than the entire object model, and so their projected shape tends to be relatively unaffected by global distortions. Thus, by performing the search on a part-by-part basis, we attain much of the robust, qualitative descriptive capabilities of, for example, Konderink's catalogue of characteristic views.

5.2 Detailed Recognition Strategy

In the learning examples reported above, there was no need to incorporate a *priori* information; it was all done by strictly bottom-up processing. However in the case at hand, the primary concern is how to integrate top-down, a *priori* knowledge of vehicle position, sensor characteristics, and object structure with sensor data to obtain the best possible estimates of object location and structure.

The approach we have taken is to model uncertainties concerning our a *priori* knowledge by using the first and second moments of the distribution of errors. That is, we characterize

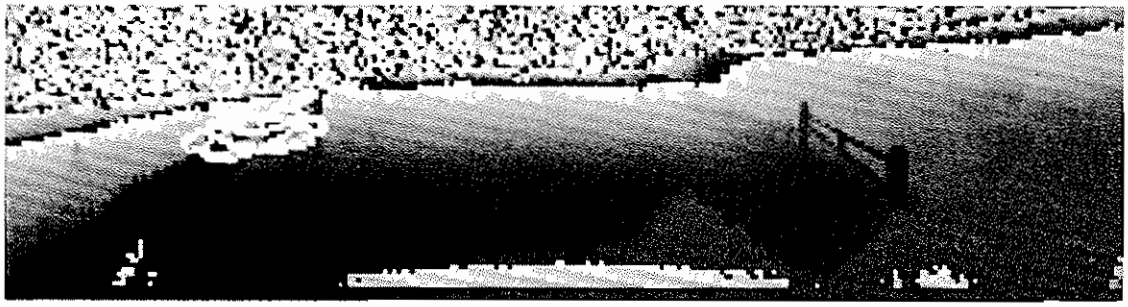


Figure 9: Prediction of gate position with error ellipses projected onto actual ALV data.

our knowledge by use of mean, variance and covariance statistics. Our choice of this representation for uncertainty is motivated by the availability of clear, simple combination rules, as well as by the small amount of information required to characterize the uncertainty. (See references [29,30] for more details regarding this approach to modeling uncertainty.)

Adopting this representation for uncertainty, our overall recognition strategy can be stated as follows:

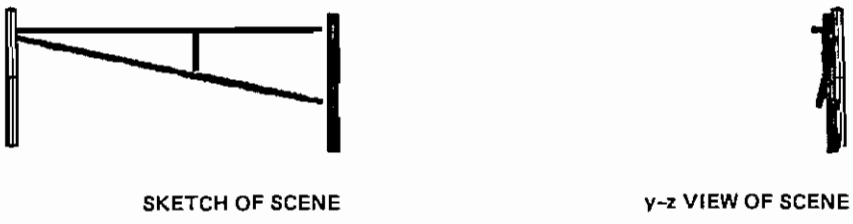
5.2.1 Prediction of Model Position

Our first step is to use the ALV's internal navigation system and our knowledge of object position, together with a model of the uncertainties in these data sources, to predict the mean, variance and covariance for the location of each of the object's parts in the range imagery. We use this information in three ways: (1) to select an image region within which we will search for the part; (2) to predict the most likely location of each part i; (3) to predict the variance of that prediction.

Figure 9 illustrates such predictions. It contains shows some ALV imagery with two-standard-deviation error ellipses superimposed upon the data. This figure shows the amount of internal-navigation-system error that has accumulated after approximately 100 feet of vehicle travel. See reference [26] for additional details.

5.2.2 Prediction of Range Values

We use our object and sensor models to predict both the detailed, pixel-by-pixel range values for each part, and the pixel-by-pixel variances associated with these values. Figure 10 shows a SuperSketch model of the gate shown in Figure 9, together with an intensity image of this gate model placed on a digital terrain map (DTM).



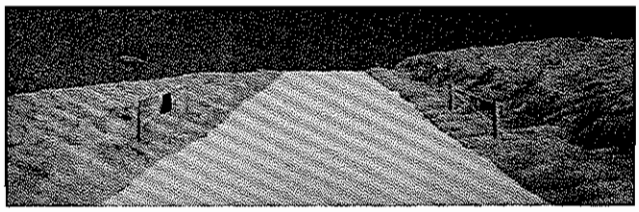


Figure 10: (a) A model of a gate constructed using SuperSketch; (b) an intensity image illustrating the gate model placed on a digital terrain map (DTM)

By combining this SuperSketch model, the DTM, and the range sensor model we can predict the range sensor values the ALV would measure at a given point in space. This is shown in Figure 11(a), along with the actual ALV range data in Figure 11(b); it can be seen that the prediction is reasonably accurate.

Finally, Figure 11(c) depicts the variance "masks" produced; these are arrays that hold the expected variance of the range predictions shown in Figure 11(b). In this illustration, light areas are predicted to have relatively low variance, while dark areas are predicted to have relatively high variance. It can be seen that pixels falling entirely on the object are predicted as being known fairly well, pixels falling on the surrounding ground are predicted as being less confidently known, and the mixed pixels along the part's edge are predicted as being very poorly known.

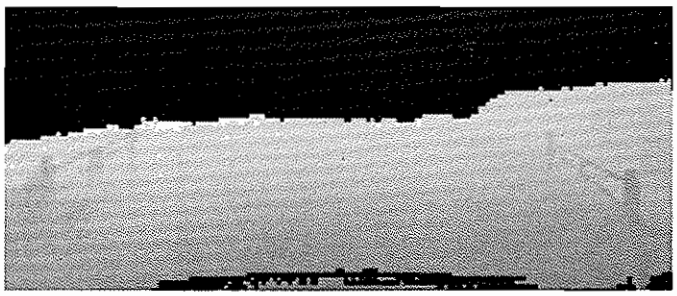
5.2.3 Computing Sums of Squares

Our objective is find the part centers (x_i, y_i, z_i) that minimize (1) the sum of the squared errors between the predicted range values and the measured range values, (2) the squared error between the final interpart center-to-center distances and the predicted interpart center-to-center distances, and (3) the squared error between the predicted part locations and the final estimated part locations.

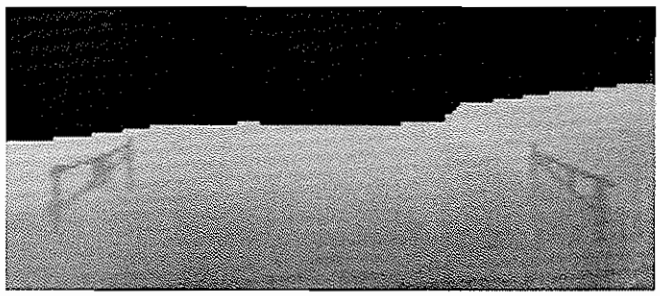
Let z_i be the predicted range at the center of part i, $\Delta z_{i,dx,dy}$ the predicted difference in range between the center of part i and the pixel displaced $(\Delta x, \Delta y)$ from (x_i, y_i, z_i), the part's imaged center. Then our predicted range values $z_{i,x,y}$ are simply

$$z_{i,x,y} = z_i + \Delta z_{i,\Delta x,\Delta y} \tag{3}$$

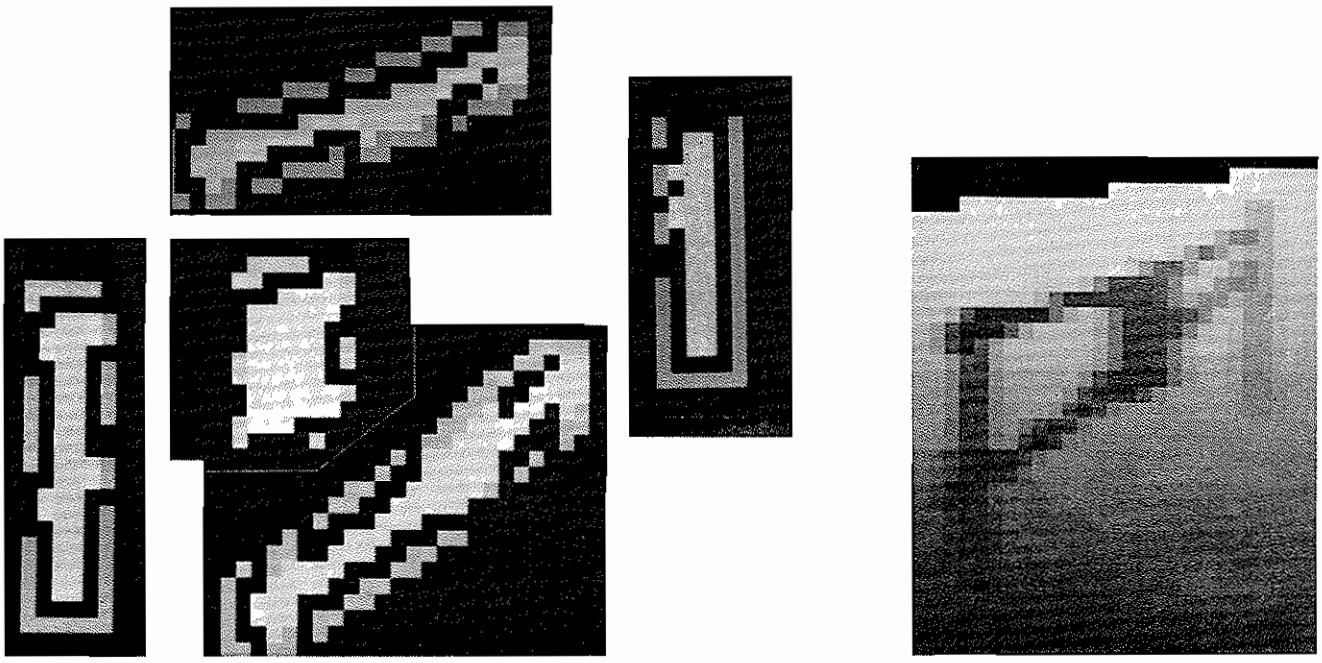
where $x = x_i + \Delta x$, $y = y_i + \Delta y$.



(a)



(b)



(c)

Figure 11: (a) ALV data of gate; (b) predicted image appearance of model; (c) "masks" showing predicted variance for pixels on and near each part; light areas have low variance, dark area have high variance.

As we change the hypothesized location of the part, our range predictions change. For a single part, however, the change is almost entirely translational; this is because single parts tend to be compact and smoothly featureless. We can thus save an enormous amount of computation at very little cost in accuracy by making the assumption that the appearance of a single part is invariant under small changes in viewpoint. This allows us to predict range values and variances once for each part, and then simply displace our predictions in x, y, and z so as to evaluate the fit between our range predictions and the measured data for each possible part location in the volume shown in Figure 9.

The first stage of our algorithm, therefore, is to calculate the squared error $s^2_{x_i,y_i,z_i}$ between the predicted range $z_{x,y}$ and the measured range $z^*_{x,y}$ when part i is centered at location (x_i, y_i, z_i):

$$s^2_{x_i,y_i,z_i} = \frac{1}{n^2} \sum_{\Delta x} \sum_{\Delta y} \left(\frac{z^*_{x_i+\Delta x, y_i+\Delta y} - z_i - \Delta z_{i,\Delta x,\Delta y}}{\sigma_{i,\Delta x,\Delta y}} \right)^2 \tag{4}$$

where $\sigma_{i,\Delta x,\Delta y}$ is our estimate of the variance of our range prediction at the pixel displaced $(\Delta x, \Delta y)$ from part i's center. Note, however, that we can "factor out" z_i because there is a unique z_i^{opt} that produces the minimum squared error for each image location (x_i, y_i). Thus we can simplify the above computation by instead finding:

$$s^2_{x_i,y_i,z_i^{opt}} = \min_{z_i} \frac{1}{n^2} \sum_{dx} \sum_{dy} \left(\frac{z^*_{x_i+\Delta x, y_i+\Delta y} - z_i - \Delta z_{i,\Delta x,\Delta y}}{\sigma_{i,\Delta x,\Delta y}} \right)^2 \tag{5}$$

We can calculate $s^2_{x_i,y_i,z_i}$ from $s^2_{x_i,y_i,z_i^{opt}}$, where $z_i = z_i^* + k$, as follows:

$$s^2_{x_i,y_i,z_i} = s^2_{x_i,y_i,z_i^{opt}} + \frac{1}{n^2} \sum_{\Delta x} \sum_{\Delta y} (k/\sigma_{i,\Delta x,\Delta y})^2 \tag{6}$$

That is, we search the measured range data for each part, computing the sum of squared errors between the predicted range values and actual values for each possible location of each part, at each point assuming the part is centered at distance z_i^{opt}, which is the best possible value for z_i.

5.2.4 Combining to Form Minimum-Mean-Square-Error Estimates

The final stage of our algorithm is to combine the mean square error between predicted and measured data values with mean square errors for part position and interpart distances to obtain a final estimate of object location and geometry.

To do this, we first calculate the interpart distance errors and the part position errors. Let $d_{i,j}$ the predicted distance between the center of part i and the center of part j, and $d^*_{i,j}$ be the distance between the current estimates of center positions for parts i and j (i.e., the "measured" distance). Then we find part centers (x_i, y_i, z_i) such that

$$\epsilon = \min_{\substack{all \\ part \\ centers}} \left[\sum_i s^2_{x_i,y_i,z_i} + \sum_i \sum_j \left(\frac{d_{i,j} - d^*_{i,j}}{\sigma_{d_{i,j}}} \right)^2 + \sum_i \left(\frac{z_i - z_i^*}{\sigma_{z_i}} \right)^2 \right] \tag{7}$$

Currently we perform this final optimization by a pruned combinatorial search; for typical examples, this search takes only a few dozen seconds on a Symbolics 3600.

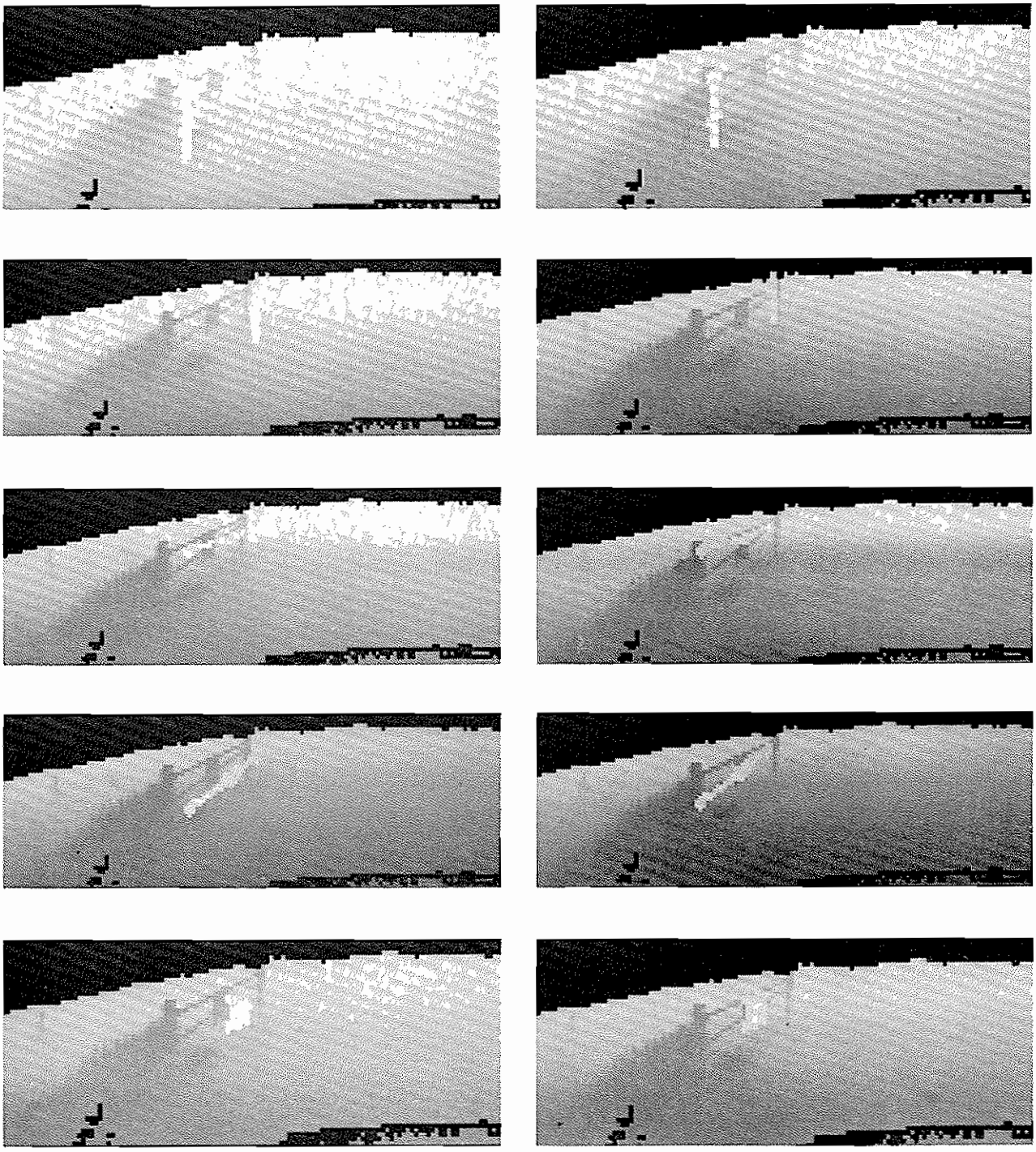


Figure 12: (a) Original estimates of part location; (b) Final estimates of part location.

5.3 Results

One method of testing the robustness of our recognition technique is into introduce errors in the a *priori* information used for prediction, and then to observe the effect of these errors on our subsequent recognition accuracy. Thus, in the following example we have intentionally included errors in the gate's position and geometry that are near the limit of what we expect to occur operationally.

Figure 12(a) shows the consequences of these errors for the predicted location and orientation of each of the gate's component parts. In this example, each part's center is displaced by about five feet horizontally and one foot vertically; moreover, the angles of the gate's cross-bars are off by about fifteen degrees, so that the further gatepost is positioned several feet lower than the nearer one. Because of the low viewing angle, the predicted range values for the ground are even more seriously wrong, averaging almost twenty feet in error. Note that because of the wide-angle-lens character of the sensor, each of these errors can result in large (15 - 20 pixel) changes in imaged position.

Figure 12(b) shows the results of applying our recognition algorithm to the ALV range data shown in Figure 11(a) using the (in error) predictions of Figure 12(a). In this figure the final estimated position for each of the object's parts is overlayed on the original range data. The positions of each part in Figure 12(b) should be compared to the initial estimate of each part's position shown in Figure 12(a). It can be seen that, despite substantial errors in the predicted part position and orientation, as well as severe errors in the predicted ground and road ranges, the algorithm has nicely located the gate and at the same time appropriately corrected the geometry of the gate model.

5.4 Summary

We have implemented a system that performs top-down recognition of an object by searching for its component parts, a procedure that minimizes the effects of global distortions caused by unknown relative orientation or poorly known overall object geometry. This procedure has the advantage of being very efficient, requiring only a few dozen seconds on a Symbolics 3600. Thus, when the vehicle is traveling in familiar areas, most of its computational power would be available for other tasks.

We accomplish object recognition by modeling the image appearance of its component "parts" sufficiently to obtain a useful estimate of each of their positions; we may then match the *ensemble* of these parts to estimate the object's position and orientation, as well as to update our model's interpart geometry. Because this algorithm employs only image-level matching, we believe that — as with the learning algorithm — we can also apply this technique directly to the outputs of shape-from-x and depth-from-x vision modules.

This system does not now make use of matching at the level of image features, nor of matching bottom-up-derived part descriptions to stored object descriptions. Both of these other matching strategies can be useful in many situations; adding such capabilities will only enhance to the robustness of our algorithm.

It is clear that this approach to recognition will not work when the object's structure or position is very poorly known. For such cases, we envision a hybrid of this top-down predictive approach and the bottom-up approach of the previous sections. For instance, if we are told only that "there is a gate on the left-hand side of the road," we do not possess

enough information to make detailed predictions. We may, however, use a stored model of a "typical" gate to predict (1) that there are likely to be long, thin horizontal and vertical parts, and (2) the likely image position and orientation of each of these parts. We can then use this information to restrict our bottom-up search algorithm to a reasonable range of part dimensions and orientations, thereby massively decreasing the amount of search required. This limited search will result in a set of horizontal and vertical parts that are reasonable constituents of a gate. Then, as with the above image-level recognition algorithm, we can search among these constituents for a combination of horizontal and verticals that best match our "typical gate" model.

6 SUMMARY

The goal of our DARPA Autonomous Land Vehicle (ALV) project is to identify potentially-useful landmarks, enter their descriptions into a database, and then, when traveling through the same region at some later time, search for and recognize these landmarks. This capability will make it possible to eventually accumulate a comprehensive catalog of landmarks that can be recognized and then utilized for accurate guidance of the vehicle. We have therefore developed a system that accomplished each of these tasks in a manner that, on the basis of these initial experiments, appears to be robust.

This system relies on the structuring of objects into their component "parts," using a representation that accords with psychological evidence about people's notions of part structure [13,14]. It appears that this representation facilitates the learning and recognition processes by providing the requisite descriptive power without using a large number of parameters. This property of concise description is especially important when the minimum-length-encoding approach to learning object descriptions is employed. In fact, we believe that, for such an approach to be successful, it *must* utilize a representation roughly similar to this one.

At present, our system does not make use of feature-level descriptions. The learning algorithm, for instance, builds object models from the data by means of direct image-to-part search. Similarly, our recognition algorithm predicts an object's image appearance and matches directly at the image level. Thus, although we anticipate that in a fully developed system *all* levels of representation would be present, the particular algorithms we have investigated represent only the outside information paths in the system structure depicted in Figure 1.

One reason for choosing these particular algorithms is to demonstrate that our approach is sufficiently general and robust that it will also perform adequately at both part-level and feature-level matching. The other reason for choosing to investigate image-level matching only is the similarity of range data to the output of shape-from-x and depth-from-x vision modules. This similarity leads us to believe that our techniques for learning descriptions and recognizing objects in range imagery may be directly transferable to situations in which it is local shape-from-x modules, rather than a laser-beam return, that provides us with image-like descriptions of the scene's geometry.

REFERENCES

[1] Badler, N. and Bajacsy, R., (1978) Three-dimensional representations for computer graphics and computer vision, *Computer Graphics, 12*, 153-160.

[2] Binford, T. O., (1971) Visual perception by computer, *Proceeding of the IEEE Conference on Systems and Control*, Miami, December.

[3] Agin, G. A., and Binford, T. O., (1973) Computer description of curved objects, *Proc. Am. Asso. for Artificial Intelligence '73*, pp. 629-635, Stanford, CA, August.

[4] Nevatia, R., and Binford, T.O.,(1977) Description and recognition of curved objects. *Artificial Intelligence*, 8, 1, 77-98.

[5] Bolles, B. and Haroud, R., (1986) Edge-chain analysis for object verification *IEEE Conf. on Robotics and Automation*, San Francisco, CA, April 7-10.

[6] Brooks, R., (1985) Model based 3-D interpretation of 2-D images, In *From Pixels to Predicates*, Pentland, A. (Ed.) Norwood N.J.: Ablex Publishing Co.

[7] Goad, C., (1985) A fast model-based vision system, In *From Pixels to Predicates*, Pentland, A. (Ed.) Norwood N.J.: Ablex Publishing Co.

[8] Faugeras, O.D., Hebert, M., Pauchon, E., Ponce, J., (1984) Object representation, identification, and positioning from range data, *Robotics Research: The First Symposium*, (Brady, M., and Paul, R., Ed., MIT Press

[9] Grimson, W.E.L., and Lozano-Perez, T. (1985) Recognition and localization of overlapping parts from sparse data in two and three dimensions, *Proc. IEEE Robotics Conference*, pp. 140-150, St. Louis, MO.

[10] Hoffman, D., and Richards, W., (1985) Parts of recognition, In *From Pixels to Predicates*, Pentland, A. (Ed.) Norwood, N.J.: Ablex Publishing Co.

[11] Leyton, M. (1984) Perceptual organization as nested control. *Biological Cybernetics 51*, pp. 141-153.

[12] Beiderman, I., (1985) Human image understanding: recent research and a theory, *Computer Vision, Graphics and Image Processing*, Vol 32, No. 1, pp. 29-73.

[13] Pentland, A. (1987) Towards an ideal 3-D CAD system, *SPIE Conf. on Machine Vision and the Man-Machine Interface*, Jan. 12-16, San Deigo, CA. Order No. 758-20.

[14]. Pentland, A. Perceptual Organization and the Representation of Natural Form, *Artificial Intelligence Journal*, February 1986. Vol. 28, No. 2, pp. 1-38.

[15] Teversky. B and Hemenway K., (1984) Objects, parts and categories, *J. Exp. Psychol. Gen.*, 113, 169-193.

[16] Rosch, E. (1973) On the internal structure of perceptual and semantic categories. In *Cognitive Development and the Acquisition of Language*. Moore, T.E. (Ed.) New York: Academic Press.

[17] Hayes, P. (1985) The second naive physics manifesto, In *Formal Theories of the Commonsense World*, Hobbes, J. and Moore, R. (Ed.), Norwood, N.J.: Ablex

[18] Thompson, D'Arcy, (1942) *On Growth and Form*,, 2d Ed., Cambridge, England: The University Press.

[19] Stevens, Peter S., (1974) *Patterns In Nature*, Boston: Atlantic-Little, Brown Books.

[20] Smith, A. R., (1984) Plants, fractals and formal languages. In Computer Graphics 18, No. 3, 1-11.

[21] Mandelbrot, B. B., (1982) *The Fractal Geometry of Nature*, San Francisco: Freeman.

[22] Gardiner, M. (1965) The superellipse: a curve that lies between the ellipse and the rectangle, *Scientific American*, September 1965.

[23] Barr, A., (1981) Superquadrics and angle-preserving transformations, *IEEE Computer Graphics and Application*, 1, 1-20

[24] Fischler, M., and Bolles, R., (1981) Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, (6), pp. 381-395.

[25] Fischler, M. *et al*, (1986) Knowledge-based vision techniques for the autonomous land vehicle program, SRI Project Report 8388.

[26] Barnard. S, Bolles. R, Marrimont. D and Pentland. A, Multiple Representations for Mobile Robot Vision, *SPIE Cambridge Symposium on Optical and Optoelectronic Engineering*, October 26-31, 1986, Cambridge, MA. Available SPIE Proceedings, Vol. 727

[27] Pentland, A. (1984a), Fractal-based description of natural scenes, *IEEE Pattern Analysis and Machine Intelligence, 6*, 6, 661-674.

[28] Pentland, A. On Describing Complex Surfaces, *Image and Vision Computing*, November 1985, Vol. 3, No. 4 pp. 153-162.

[29] Donald F. Morrison, *Multivariate Statistical Methods*, McGraw-Hill Inc., New York, N.Y., 1967.

[30] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill Inc., New York, N.Y., 1965.

[31] Bajcsy, R. and Solina, F., Three-dimensional Object Representation Revisited, *First International Conf. on Computer Vision, '87*, June 8-11, London, England.

[32] Smith, A. R., (1984) Plants, fractals and formal languages. In Computer Graphics 18, No. 3, 1-11.

[33] Reeves, W. T., (1983) Particle systems - a technique for modeling a class of fuzzy objects, *ACM Transactions on Graphics 2*, 2, 91-108.

[34] Saches, Oliver. *The Man Who Mistook His Wife For A Hat*, Boston: M.I.T. Press.

[35] Winston, P.H., (1975) Learning structural descriptions from examples, Ph.D. Thesis, in *The Psychology of Computer Vision*, Winston, P.H. (Ed.), New York: McGraw-Hill.

[36] Winston, P., Binford, T., Katz, B., and Lowry, M. (1983) *Proceedings of the National Conference on Artificial Intelligence (AAAI-83)*, pp. 433-439, Washington, D.C., August 22-26.