RECOGNITION BY PARTS

Technical Note No. 406

December 16, 1986 (revised August 25, 1987)

By: Alex P. Pentland

Artificial Intelligence Center
Computer and Information Sciences Division

Approved for public release; distribution unlimited

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

RECOGNITION BY PARTS

Alex P. Pentland

Artificial Intelligence Center, SRI International
Menlo Park, California
and
Center for the Study of Language and Information
Stanford University

ABSTRACT [1]

We argue that most natural objects have a *part structure* that we can recover from image data and thus use as the basis for "general-purpose" recognition. We describe a "parts" representation that is fairly general purpose, despite having only a small number of parameters. Having this expressive power captured by a small number of parameters allows us to approach the problem of recovering an object's part structure by use of minimal length encoding. We present several examples of recovering part structure using various types of range imagery to show that the recovery procedure is robust.

1 THE PROBLEM

To have a general-purpose machine vision capability, we must be able to *recognize* things. This involves computing a sufficiently canonical description of the objects in our environment that we can match these recovered descriptions to stored descriptions. Moreover, we must be able to learn new object descriptions; thus we must be able to compute a canonical description of an object without much reliance on previously-learned descriptions.

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 people are somehow recovering this "natural" structure from imagery and using 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 will, therefore, argue that most natural objects have a *part structure* that we can recover from image data, and that is exactly this structure that provides the computational basis for both recognition and learning.

This is not to say that lower level image features do not contribute to recognition. We argue, however, that the way in which they do so is by helping to define an object's part structure. It is clear, for instance, that most preattentive image features — e.g., T-junctions, parallel lines, and the like — strongly constrain part structure [1-6]. Moreover, it seems to us that image features *alone* cannot generally support recognition. There is, for instance, a medical literature concerning patients whose spatial and feature-recognition abilities remain

[1] This research was made possible by National Science Foundation grant no. DCR-83-12766, by Defense Advanced Research Projects Agency contract no. MDA903-86-C-0084, and by a grant from the Systems Development Foundation. I wish to especially thank Marty Fischler, Yvan Leclerc, and Bob Bolles for their comments, help, and insight.

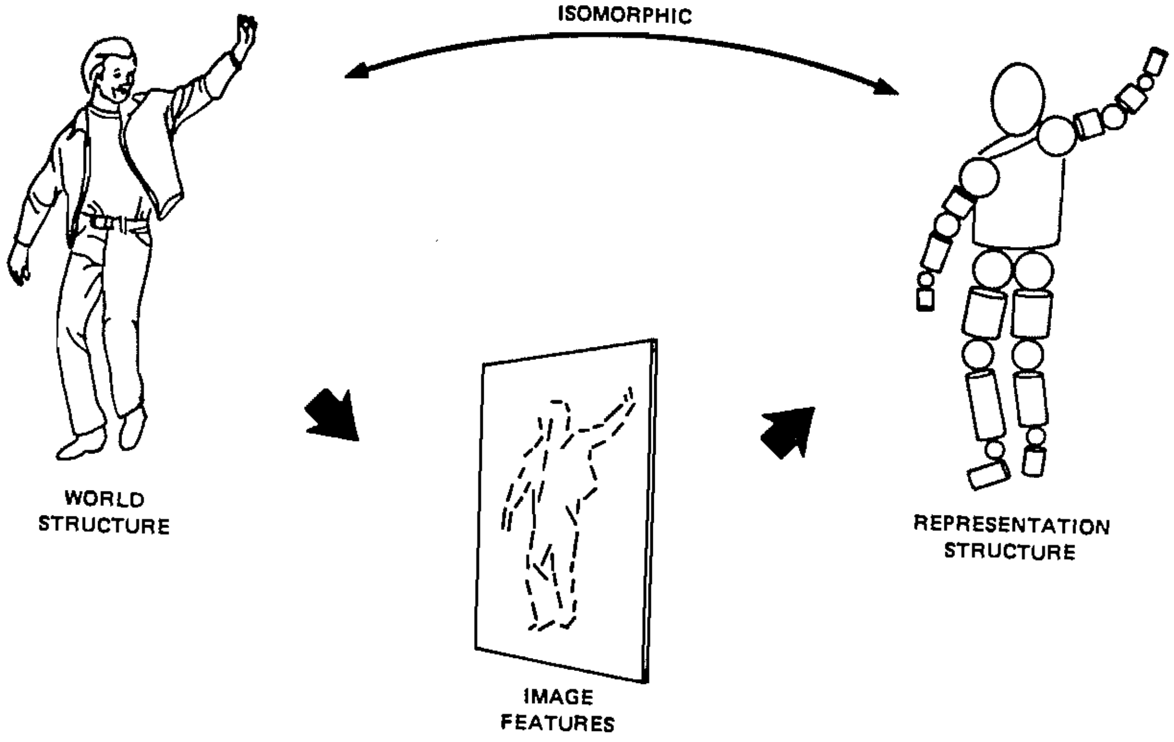

Figure 1: Our view of the process of vision: World Structure ↦ Image Features ↦ Part Representation. General-purpose recognition requires an isomorphic relationship between the viewed object's true 3-D part structure and the structure of our representation of it.

intact, but who are unable to recognize objects except in very special, highly constraining situations [7].

Thus our view of the vision process is illustrated in Figure 1; it starts with image features and proceeds by computing a representation that is isomorphic to both the object's 3-D metric properties (e.g., shape) and to the 3-D *part structure* of the object. We see this part-by-part isomorphism between internal representational structure and objective causal structure as being the basic requirement for general-purpose recognition; only if we can obtain such a part-by-part correspondence can we match between the stored and newly-computed representations.

It is difficult to accurately define this notion of "part structure" We know, for instance, that it is related to rigidity, and that things that move separately must be separate parts. Similarly, part structure is related to object dynamics. In mechanical design, for instance, complex objects are standardly broken down into "parts" in order to analyze their dynamic and kinematic properties. Finally, however, we must fall back on the fact that people seem to have strong, consistent intuitions about "part structure." The particulars of the representation we employ in this paper were originally determined by use of psychological evidence about people's notions of part structure [8].

Although it is difficult to define the details of such a "part structure" theory, it is relatively easy lay out requirements that such a theory must satisfy if it is to be useful for recognition. This leads us to a list of requirements that is similar to — but in some critical ways different from — that compiled by Marr and Nishihara [9]:

(1) The representation must have *descriptive adequacy,* i.e., it must be sufficient to describe the image data in a natural manner, one that captures the structural distinctions needed by subsequent reasoning processes. However it does *not* seem necessary, or even desirable, that a part representation account for every detail of the image data. People, for instance, often do not notice small differences between generally similar photographs. It seems sufficient for our part representation to be able to represent the data at a "cartoon-like" level of detail, i.e., to capture (roughly) the amount of detailed shape information that people typically remember. We can later employ a separate surface representation, such as a thin-plate surface model [10] or a fractal surface model [11], in order to describe the [relatively small] differences between our part representation and the actual detailed surface shape.

(2) The recovered description must be *stable*, so that we can use it to match against stored descriptions. There seem to be three important types of stability:

- Stability with respect to scale: We require not only insensitivity to small changes in scale, but the also ability to "summarize" small details appropriately, e.g., given a large image of a person to separately describe ears, nose, etc., but that given a very small image of a person to describe the head as a single entity. This is similar to the Marr and Nishihara suggestions [9] concerning the need for a hierarchical, multiscale representation.

- Stability with respect to image noise: The ability to produce similar descriptions from both accurate, clean data and from noisy, or even silhouette, data.

- Stability with respect to configuration: The ability to produce similar descriptions as the viewer's position changes, and as objects articulate. Because recognition requires matching stored descriptions against recovered ones, the ability to produce a stable part-structure *segmentation* is critical to the success of our approach. An unsegmented surface representation, for instance, generally can not support such matching because even small variations in surface shape or viewing position typically changes the number of spline knots, polygons, or surface patches, and thus prevents symbolic matching.[2]

In short, these requirements demand a representation that is general-purpose enough to describe the situation in a natural manner (i.e., one that captures distinctions needed by later reasoning processes), and a recovery process that produces sufficiently canonical descriptions that we can match the recovered description to our stored models.

The plan of this paper, then, will be to briefly present a theory and representation of part structure, describe a method for computing that part structure from image data, and then evaluate the combination of recovery method and representation against the above criteria.

[2] When using a surface representation matching between two surface representations is almost always accomplished by using each of the surface representations to render a synthetic image, and then calculating the pixel-by-pixel RMS error between these two synthetic images. Thus in a certain sense an unsegmented surface representation is actually worse than raw images for object matching and recognition!

2 REPRESENTATION

Many modern psychologists [4-6], as well as the psychologists of the classic Gestalt movement, have argued that we conceive of the world in terms of *parts*, and that the first stages of human perception are primarily concerned with structuring the image into these "parts." This part-structure is seen as forming the building blocks upon which we build the rest of our perceptual interpretation.

One reason people may favor such part descriptions is that they offer considerable potential for reasoning tasks. It seems, for instance, that people employ such descriptions in commonsense reasoning, learning, and analogical reasoning [12-14]. 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 between parts of objects, rather than of local surface patches or of particular instances of objects.

Moreover, recent research in graphics, biology, and physics has provided us with 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 [15-18]; i.e., that our world can be modeled as a relatively small set of generic processes — bending, twisting, 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.

Following in this tradition, we have explored [8] how people describe shape using both the tools of protocol analysis, and the psychophysical method devised by Triesman. One concise characterization of our results is that we found our subjects describing 3-D shape *procedurally*: by describing how one would make the shape using a malleable material such as clay, using a few generic forming actions [19]. As an example, our subjects might have described the back of a chair is a rounded, flattened 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).

In our experiment it did *not* seem that people tried to describe the exact surface shape. Rather, they appeared to be trying to describe the general, "global" structure of the form, with the detailed surface shape being described (where necessary) as variations from the overall shape. For instance, a person's forearm would be described in two parts: *globally* as a tapered cylinder, and *locally* by small deviations from the global form which describe the detailed musculature. Thus this sort of parts-level description should be thought of as being in addition to, for example, thin-plate or fractal surface models. It is the frame upon which hangs detailed descriptions of surface shape.

2.1 Our Representation of Part Structure

Inspired by how people seem to describe shape, we have adopted a representation that describes shape in a similar manner, e.g., 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 deformed and shaped, but which is intended to correspond roughly to our naive perceptual notion of "a part." For this

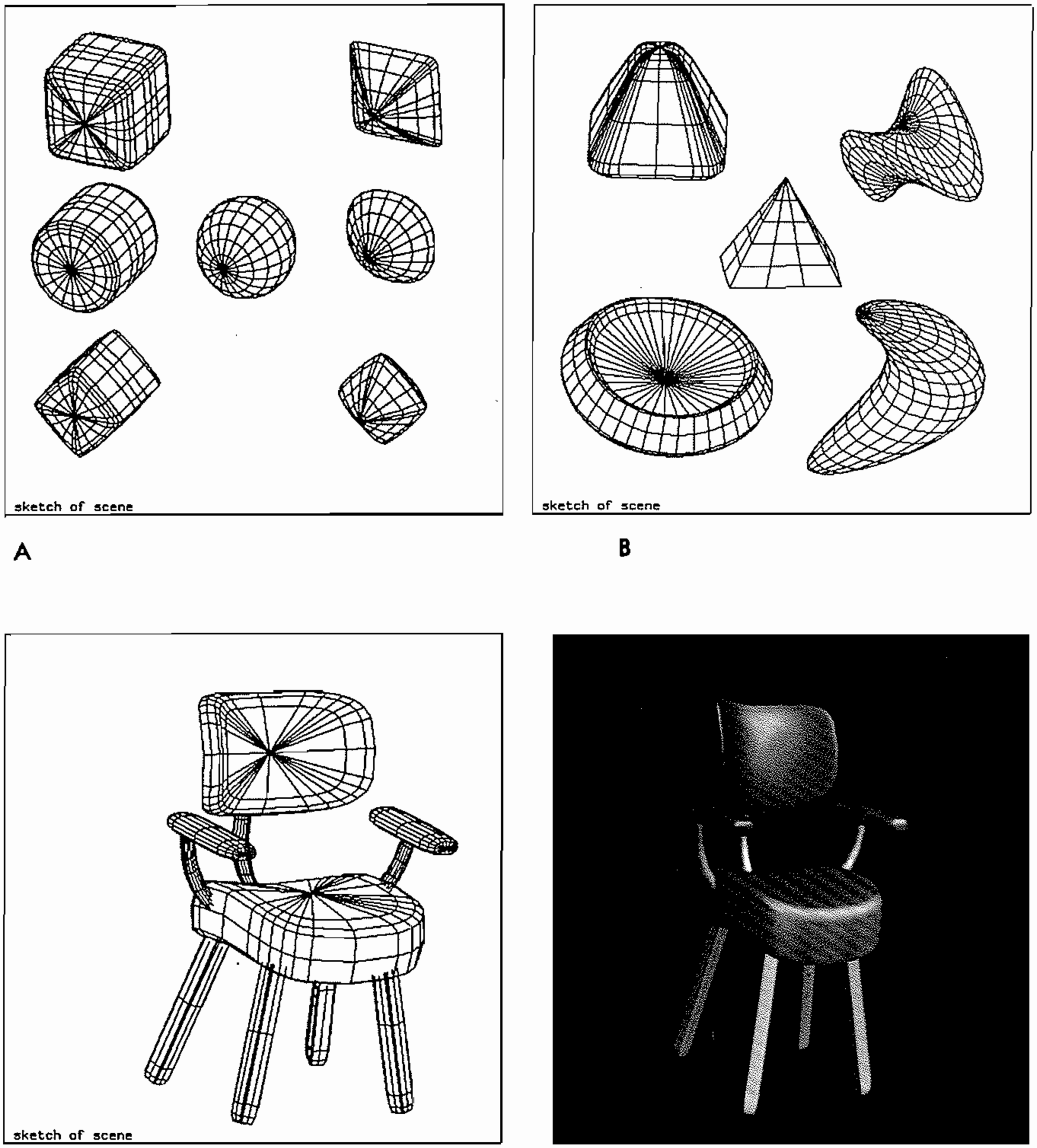

Figure 2: (a) A sampling of the basic forms allowed, (b) deformations of these forms, (c) a chair formed from Boolean combinations of appropriately deformed superquadrics.

basic modeling element we use a parameterized family of shapes known as a *superquadrics* [20,21], invented by Danish designer Peit Hein, which 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 constructive solid geometery (CSG) modeling primitives currently in common use.

These basic "lumps of clay" are used as prototypes that are then deformed by linear stretching and tapering, or quadratic bending, and then combined using Boolean operations to form new, complex shapes.[3] Our representation, therefore, is both a generalization of the CSG approach and a modification of the generalized cylinders approach; we are combining a restricted class of generalized cylinders using Boolean operations.

We have constructed a 3-D modeling system called "SuperSketch" that employs this shape representation. This real-time, interactive modeling system is implemented on the Symbolics 3600, and allows users to interactively create "lumps," change their squareness/roundness, stretch, bend, and taper them, and finally to combine them using Boolean operations. This system was used to make the images in this paper, and, by writing to the author, is available free to colleges and universities.

The specification of shape, orientation, position, and the various deformations requires a total for 14 parameters.[4]. This compares favorably with the nine parameters needed to describe intensity, first, and second derivatives at a single point; the nine parameters needed simply to describe the position, orientation, and size of a rectangular solid; or the hundreds of parameters that might be needed to describe a generalized cylinder.

We have found that this representational system has a surprising generative power that allows the creation of a wide variety of form, such as is illustrated by Figure 2. As a consequence of the concise nature of this shape language, we have found that even complex objects and scenes can typically be modeled using a relatively small number of parameters, as is illustrated by the models shown in Figure 3. These modeling results have lead us to believe that this representation provides a concise "natural" level of description, of exactly the sort needed to support recognition.

2.1.1 Local versus global representation of shape

As discussed above, this parameterized shape vocabulary is *not* intended to describe the surface shape in full detail. Rather, our lump-of-clay specification is a coarse-grain descrip-

[3] See the Appendix for more mathematical details of the representation.

[4] 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 come from our finding that additional degrees of freedom were almost never used when people constructed models using SuperSketch.

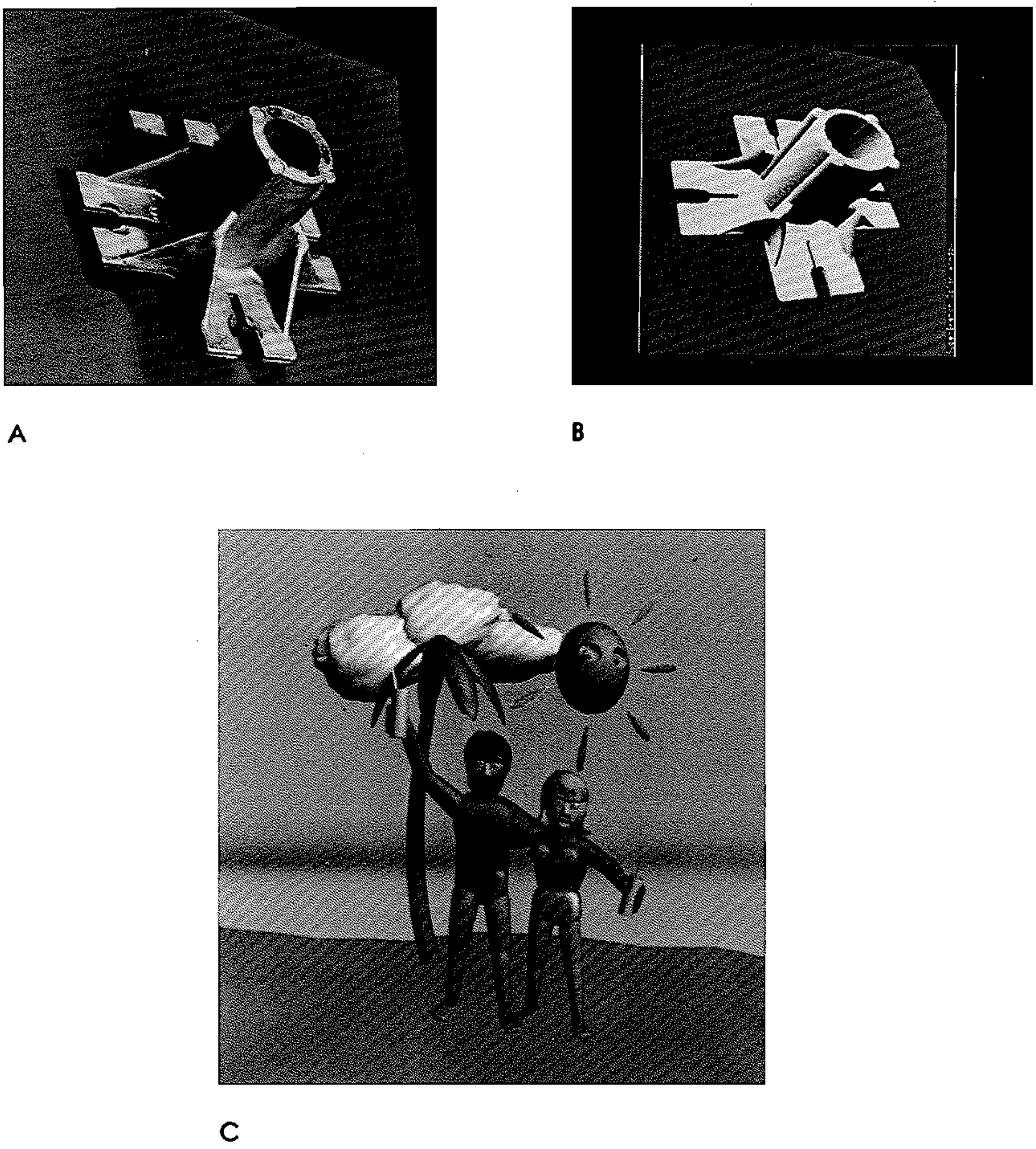

Figure 3: (a) An industrial casting, (b) Its SuperSketch model (approximately 300 parameters; construction time: 23 minutes), (c) A SuperSketch model of two people (approximately 1000 parameters; construction time: approximately 4 hours)

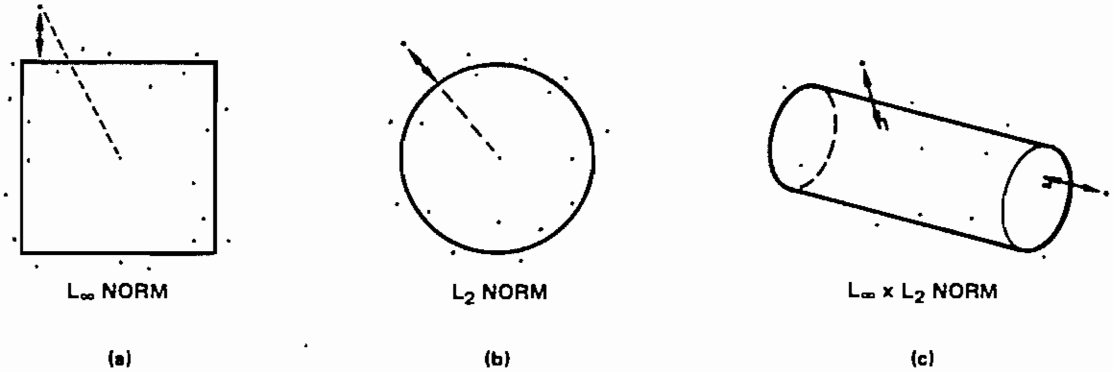

Figure 4: Global shape is important in integrating positional information. (a) Measuring distance between a point and the center of a cube, (b) a point and a sphere, and (c) a point and a cylinder.

tion of the general structure of the form. Having captured a first-order approximation to the overall form by use of our shape language, we can now describe the details of the surface shape as small deviations normal to the superquadric's surface.

There are three major advantages to describing shape by use of a combination of a global parameterized model together with a point-by-point surface model. First, separating our description into global and local representations nicely supports the differing requirements of various reasoning tasks. For instance, in calculating an object's dynamics or kinematics we need only know that the object is approximately cylindrical, while determining a gripping point requires knowing about local deviations from the overall cylindrical shape. Second, such a global-local representation allows a much more concise encoding of the detailed surface shape, because the larger-scale variations in the form are succinctly encoded by our simple parameterized shape vocabulary.

And finally, having a global description is necessary to correctly integrate positional information when measurement errors are normal to the surface rather than parallel to the viewer, as is illustrated by Figure 4. Standardly such fitting is accomplished by projecting data points along the shape's field of surface normals, calculating the point of intersection, and minimize the sum of Euclidian L_2 distances between data point and intersection.

An computationally simpler approach to such fitting is to tailor the distance metric to the problem. For instance, the problem of fitting a cube to a set of position measurements can be easily easily accomplished by simply minimizing the sum of the L_∞ ("maximum") norm distances between the measured points and the center of the cube, with no intersection calculation or projection being required. Similarly fitting points to a cylinder is easily accomplished using an orthogonal combination of the L_∞ and L_2 norms. The general form of this relationship between shape and distance metric is the following. If we define a shape by

$$D = \left(((\Delta x)^\alpha + (\Delta y)^\alpha)^{\beta/\alpha} + (\Delta z)^\beta\right)^{1/\beta} \tag{2}$$

with[5] $D = 1$ then D is the computationally simplest distance metric (i.e., the $L_\alpha \times L_\beta$ norm) when measurement error is assumed normal to the surface. In particular, the distance metric associated with a particular superquadric shape is precisely it's standard inside-outside function.[6] See, for example, reference [22].

3 PART RECOVERY: MINIMAL-LENGTH ENCODING

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 and so we will use range imagery exclusively in this paper.

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 from ground is described in [23]; 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 comprising a single figure, and having adopted a particular representation, we can now pose the problem of learning an object description as the process of using our 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.

3.1 Our Optimization Criterion: Minimal-Length Encoding

We shall employ encoding length as our means of evaluating an explanation of the image data, in other words, our goodness-of-fit criterion is the cost (in bits) of encoding the image data by use of our 3-D shape vocabulary and a simple noise model. The total cost of encoding some image data will be calculated as being the sum of several subterms; these are the cost of specifying: (a) the parts description itself, (b) the "noise" in the data (small deviations from the value predicted by the parts description), (c) errors of commission (points predicted that don't occur in the data), and (d) errors of omission (image points missed by the parts description). Thus the "best" explanation of some image data we will take to be that encoding (in terms of part models and noise) that has the minimum sum of costs (a) through (d).

There are two motivations for adopting the minimal-length encoding approach. First, finding the minimal-length encoding of a scene is equivalent to finding the maximum a *posteriori* (MAP) explanation of the scene, where the assumed encoding costs are inversely proportional to the log of the prior probabilities. Thus by finding a minimal-length encoding of image data using our vocabulary, we actually recover the MAP estimate of the both the large-scale image structure, which we will model by use of our parts vocabulary, and the small-scale image structure, which we will model as point-wise independent shot noise in the range sensor's distance measurement.

[5] A sphere has $\alpha = \beta = 2$, a cube $\alpha = \beta = \infty$, and a cylinder $\alpha = 2, \beta = \infty$.

[6] Further, when we extend our basic shapes to include bent, stretched or tapered forms the inside-outside function remains the appropriate distance metric.

Second, if we know the generative structure of our scene — if we know about the processes that create the world we are observing — then the minimal-length encoding of the scene in terms of those processes is the simplest explanation of the scene's causal history. This technique is exactly what geologists or paleontologists use when reconstructing the folding of rock strata or development of a family of animal species. Such application of minimal-length encoding is the formal version of "Occam's Razor:" the principle that the simplest explanation is the best explanation [7].

Our protocol analysis research indicates that people tend to think of the world in terms of formative processes that are analogous to sculpting in clay. If the world *really is* formed of processes analogous to our vocabulary of construction and deformation, then by use of minimal-length encoding we can use the image data to infer a description that can be related to the scene's causal structure, and thus to its functional significance.

It is important to understand that that the formative processes in the scene do not have to actually *be* clay sculpting in order to determine causal structure: They only have to be *isomorphic* to the clay-sculpting operations. For instance, the growth of a plant's stem is isomorphic to the stretching and tapering of a cylindrical primitive, it's branching is isomorphic to the Boolean combination of cylindrical primitives, and it's curving toward the sun is isomorphic to our bending operation. Thus if we can recover a description of that plant in terms of our shape vocabulary, our description will be isomorphic to the "true" explanation of the plant's growth. As a result, the distinctions our description makes will in fact be distinctions with functional significance.

Thus the most important requirement when employing a minimal-length encoding approach is to use a shape vocabulary that is isomorphic to the actual 3-D structure of the scene. If we choose our vocabulary correctly, then a minimal-length encoding using that vocabulary will provide us with a meaningful segmentation of the data. This fact provided the motivation for our choice of a shape vocabulary based upon psychological evidence concerning people's notions about the intrinsic structure of 3-D objects [4-6,8,12-14,19].

The primary difficulty in computing a minimal-length encoding is that it requires global optimization of the cost 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.

3.2 Decomposability of Our Optimization Problem

The first property we may take advantage of is that we may decompose of our search for the best explanation 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.

In the minimal-length encoding framework we assume that that the scene is in fact generated by our shape vocabulary, and that noise was then added to the image data. Thus

[7] That is, one can prove that if a body of data is generated by a 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. Thus one can formally define the intuition that the "simplest explanation" is in fact the correct one.

locality of effect in the minimal-length encoding follows from the locality of our assumed generation process: When we add a new part to a scene, it does not affect the description or appearance of parts beyond it's imaged boundaries. Thus, similarly, changes that are far enough from a particular image point can 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 much farther than 50 pixels to find the part model that provides the best description of the image region surrounding that point.

Our optimization procedure, therefore, will first determine the best localized encoding using a *single* SuperSketch part descriptor, taking into account the cost of "noise" errors and of errors of commission. The cost of errors of omission can not be evaluated locally, because image data left unaccounted for by one part descriptor may later be accounted for by some other part. These localized "best" encodings will then subsequently be combined into a *global* minimal-length encoding.

In making use of this locality property, we have implicitly ruled out the use of global descriptors such as symmetry or repetition (descriptors that specify distribution of the parts) as participants in the very first level of perceptual organization. We are asserting that such meta-level descriptors must operate upon already-discovered local structure; that is, *first* you find the range of possible local explanations and *then* you search among all of the local descriptions for the best global explanation, taking into account properties such as symmetry, parallelism, and repetition.

3.3 Finding the Localized Minimal-Length Encoding

We are still left with the difficult problem of finding the best fit of a single SuperSketch part model to a region of image data. To solve this problem we will make use of the fact that this fitting problem is generically well-behaved: It has a broadly-tuned, well-defined, and stable solution.

Such non-linear optimization problems are frequent in vision. Typically solution is attempted by use of either a variant on gradient ascent, or by use of simulated annealing [24]. Unfortunately, all gradient techniques fail when local maxima are present, and simulated annealing is too slow to be practical.[8]

Recently several authors have suggested using continuation methods (e.g., scale space) to solve global optimization problems [25]. But except for very special cases, where the continuation makes the problem convex, these methods are still foiled by local maxima. Alternatively, simulated annealing with an accelerated cooling schedule has been suggested as a method of global optimization, however use of such a schedule means that convergence is no longer assured.[9]

We observe, however, that despite these grim facts most researchers routinely use gradient ascent to find reasonable solutions [26]. This is because most physically-motivated systems are *generically well-behaved*, that is, if we examine the equations that describe a physical system we find that over most of the parameter space the system varies both

[8] To achieve convergence simulated annealing requires maintaining statistical equilibrium; this requirement means that global search of the entire state space is often a more efficient strategy!

[9] Use of such accelerated cooling schedules results in an algorithm that is best described as gradient ascent/descent with Poisson noise added to the energy function; the addition of noise makes the algorithm somewhat more resistant to being trapped by shallow local maxima, but in no way assures success.

smoothly and slowly. It is this generic behavior that allows gradient ascent techniques to work as well as they do: Most of the time physical systems are convex in a broad neighborhood surrounding the global maximum. Thus it is often the case that if we have a rough estimate of the correct parameter settings then we can use gradient ascent to find the global maximum.

3.3.1 Massively parallel search

We can exploit this generic behavior to devise a global optimization procedure; an example will illustrate our point nicely. Let us suppose that we have a system of equations for which we must find the global maximum, and that this system of equations has six variables or parameters. Let us further suppose that the convex region surrounding the global maximum spans roughly 1/3 of the range of each of these parameters. Thus the convex region surrounding the global maximum occupies roughly $1/729^{th}$ of the total area of the entire parameter space.

An example of such a problem is minimizing:

$$f(\vec{X}) = \prod_{i=1}^{i=10} a_i \cos(1.5\pi x_i + \theta_i), \qquad 0 \leq x_i \leq 1 \tag{3}$$

with respect to $\vec{X} =< x_i >$, where the a_i and θ_i are unknown constants. This is clearly a problem that will be impossible for gradient ascent, and difficult for simulated annealing.

However if we sample the function at each of $6^6 = 46,656$ points evenly spaced throughout our parameter space, then we will be *guaranteed* of sampling at (at least) one point where each of the parameters are within about 8% of best parameter value, and well within the convex region surrounding the global maximum. Further, we are guarenteed that that point will have a value that is at least half the value of the global maximum value. Consequently, only points with values more than half the maximum value discovered can possibly be near the global maximum, and so if we start gradient ascent searches from this subset of points we will be *guarenteed* of finding the global maximum. For typical values of the a_i, there will be only a very few points that are candidates for being near the global maximum.

The advantages of this approach emerge when using a massively parallel computer such as a Connection Machine. With such a machine we can in *one* parallel operation prune away most of the parameter space, leaving only a small set of points which are potentially near to the global maximum. Moreover, in cases where the global maximum value is more than three times the value of the other local maxima, our one parallel operation will produce a *single* point which is guaranteed to lie near the global maximum. [10]

Figure 5 illustrates such a case. Figure 5(a) shows a function to be maximized; note the many local maxima that will foil gradient ascent techniques. Figure 5(b) shows the same function thresholded at one-third of the maximum value; it is clear that only points near the global maximum have greater values. Thus we can use a coarse sampling of the function shown in Figure 5(a) to find — in one parallel operation — a point near the global maximum.

[10] There may also be points adjacent to this point and within the covex region surrounding the global maximum, but not points distant from it in the parameter space.

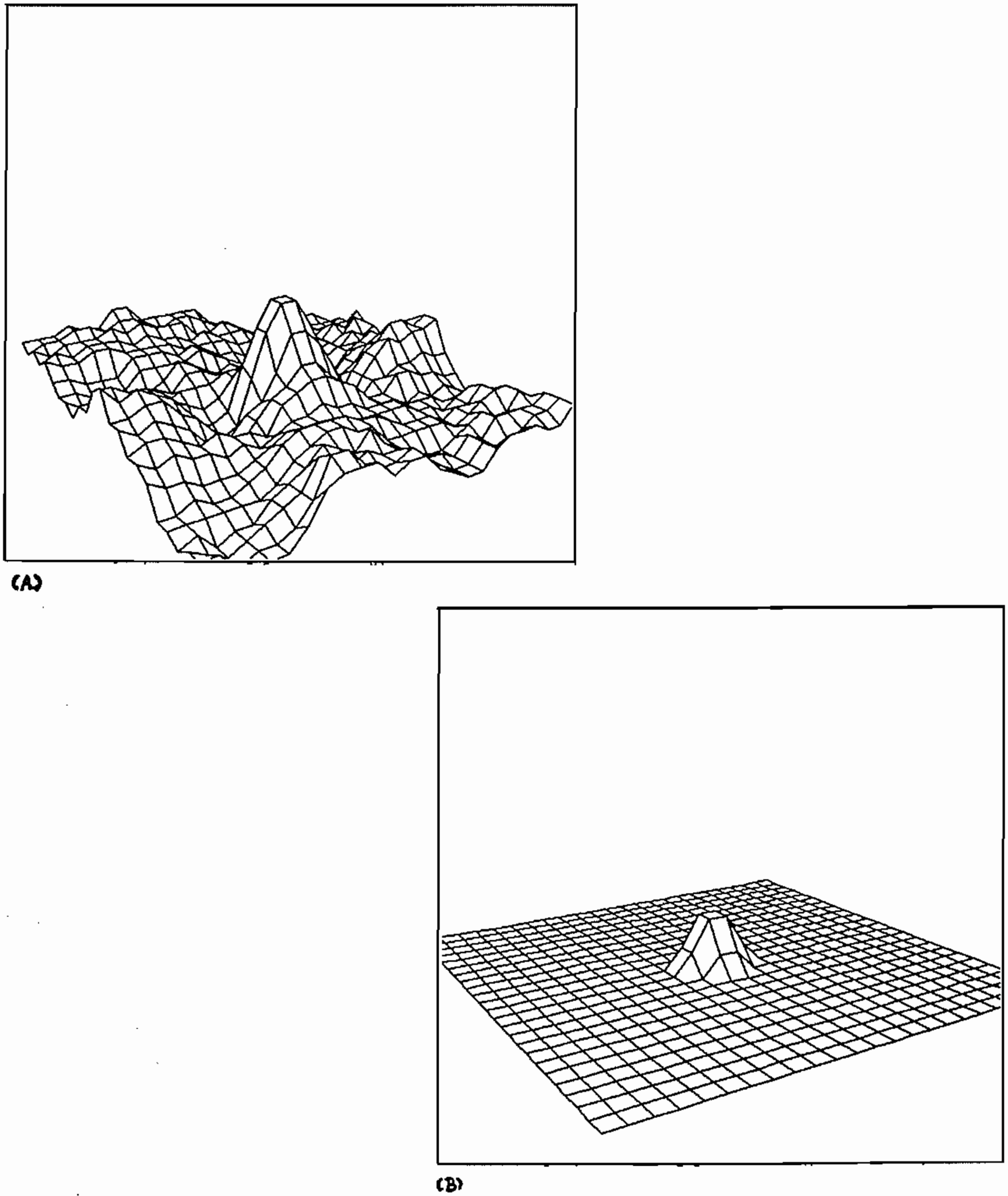

Figure 5: (a) A highly nonlinear fitting function with many local maxima, (b) the same function thresholded at one-third of the maximum value. Having a distinct, broad global maximum (such as in this example) can allow us to use coarse search to find a global optimum in a single parallel operation.

Getting away from the specifics of these examples, if we know that the two-thirds-power diameter of the convex region surrounding the global maximum is of width c_i in each of the parameters x_i, and that the range of these parameters is r_i, then we require $\Pi_i \frac{r_i}{c_i}$ sampling points. Thus the complexity of our procedure grows as fast as the size of the state space so that as the number of variables increases, the number of starting points increases dramatically. This limits practical application of this approach to problems with less than ten or twelve parameters.

3.3.2 Fitting SuperSketch models

The problem at hand — that of fitting SuperSketch parts to range data — is also susceptible to this approach: We can show that there are less than 2^8 local maxima, and that our goodness-of-fit measure is a smooth, well-behaved function of the model parameters (The attached Appendix presents this argument; see also Solina and Bajcsy's work [26] on fitting of SuperSketch-like modeling primitives). Thus we might expect that by sampling each parameter sufficiently densely we could guarentee that at least one sample point was both within the convex region surrounding the global maximum.

The intuition behind this argument is illustrated by Figure 6. Figure 6(a) illustrates the rendering of a range image of a banana-like shape, and then Figure 6(b) illustrates comparing the hypothesized range values to the measured range values in order to compute our "goodness-of-fit" measure. Figure 6(c) shows how the value of this measure varies as each of the part's parameters are varied. At the center of each graph in Figure 6(c) is the exactly matching parameter value; it can be seen that our "goodness of fit" measure varies slowly as we move away from the correct parameter value.

Figure 6, therefore, illustrates that near the optimal solution the problem of finding the best fit is broadly convex. We have used the expanding grid method described in the Appendix to ascertain that the convex region surrounding the correct parameter setting is always quite broad: For instance, length, width, or depth can be varied by up to 50% before the goodness-of-fit value falls to two-thirds of its original value. In contrast, the fit is relatively sensitive to orientation: Rotation angles can be varied by only 10 or 15 degrees before dropping to two-thirds of the optimal value.

We therefore adopted the strategy of evaluating our goodness-of-fit functional at roughly 2^{16} points in parameter space, 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) [11]. Thus no point in the parameter space differs from one of our sample points by more than an average of 12% along each parameter.

We have examined a large number of test cases to verify that this sampling rate is sufficient to ensure finding the global maximum. We found that for a typical test case a single sample point would provide a fit more than twice as good as any other, so that it could immediately be taken as the being in the neighborhood of the global maximum. In none of our test cases were there more than four points whose goodness-of-fit value was within a factor of two of the best value discovered, and in such cases each of these points

[11] 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.

provided an intuitively reasonable fit to the range data.

Thus in most of our test cases we could uniquely identify a parameter setting which was in the neighborhood of the global maximum using only one parallel evaluation [12] of our goodness-of-fit functional. In the remaining test examples this single parallel evaluation was sufficient to identify a small number of good parameter settings, one of which was in the neighborhood of the global maximum.

We have confidence, therefore, that our optimization procedure will find a part model that is close to the optimal minimal-length encoding of that region of image data (when using a single part model). By repeating this optimization at each (coarsely quantized) image position, we produce a small set of part models each of which provides the best region-by-region encoding of the image data. We can then search among combinations drawn from this set of regional best-fits to find a small set of parts that provides the best overall explanation of the image data, i.e., an approximately minimal-length encoding of the image data in terms of our modeling primitives.

3.4 Avoiding Errors Due to Occlusions

One of the key elements assuring the success of this approach is evaluation of the localized 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 a range image that correctly accounts for the effects of perspective, surface shape, and other known variables that affect image appearance. 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. Steps (1) and (2) are illustrated by Figure 6(a)

(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. 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). By using a RANSAC-style [27] histogramming procedure, we allow large portions of the figure to be occluded without disturbing the matching process. This is illustrated by Figure 6(b).

(4) From this histogram we estimate the position p of the largest peak. This peak is the most frequently occuring 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

[12] I.e., 2^{16} independent evaluations evenly distributed across the parameter space.

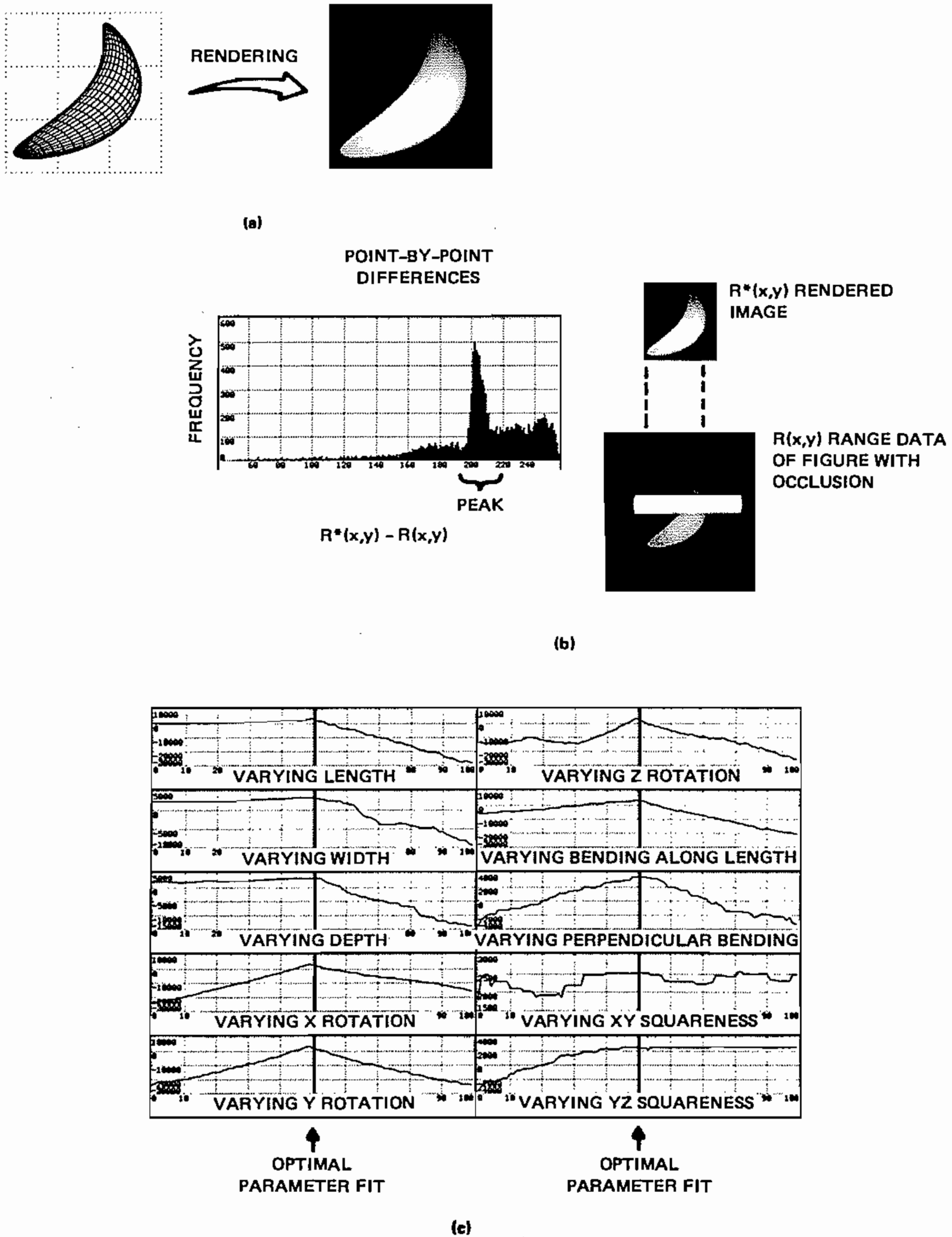

Figure 6: Finding the best fitting part descriptor. (a) We take a 3-D SuperSketch part with the hypothesized parameters (e.g., orientation, length, squareness), render it using known sensor characteristics to produce a predicted image $R^*(x,y)$. (b) We then histogram point-by-point differences (i.e., $R^*(x,y) - R(x,y)$) between the R^* and the range data $R(x,y)$; the size of the largest peak is used to calculate our goodness of fit measure. (c) 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.

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:

$$\Gamma = \mu - \lambda\epsilon \tag{4}$$

In our examples we have used $\sigma = \lambda = 4$; the procedure seems to be relatively insensitive to the value of these parameters.

The measure μ gives us the number of pixels accounted for by this hypothesized part to within $\pm 1/2\sigma$, and the measure ϵ gives us the number of pixels that are errors of commission. As a consequence the quantity

$$s = -\kappa_0 + (8 - \kappa_1)\Gamma \tag{5}$$

estimates the amount of savings (in bits) gained by encoding 8-bit range data using the hypothesized part parameters, where κ_0 is the bit cost for describing the hypothesized part (assumed constant for all parameter settings), κ_1 is $\log(\sigma)$, the average cost of encoding the difference between the range data and the hypothesized surface (data noise errors), and λ is set to the cost of errors of commission divided by $(8 - \kappa_1)$.

Thus we can determine the *localized* minimal-length encoding by maximizing Γ over all of our model parameters. Having accomplished this at each image point, we can then use these localized encodings to compute a global minimal-length encoding.

3.5 Finding the Global Minimal-Length Encoding

To determine the global minimal-length encoding, we choose from among these localized minimal encodings a subset that best describes the image. That is, we must find the set of part models that minimizes the combined cost of part description, noise errors, and errors of both commission and omission. This task is difficult because the localized part models are not independent; they overlap and obscure each, so that encoding cost for the set is not closely related to the sum of the individual encoding costs.

In general, then, we must consider n-ary relationships in order to find the global minimal-length encoding. Binary relationships, however, constitute almost all of the part-to-part interactions in calculating the minimal-length encoding, because it is relatively rare for three or more parts to overlap across a significant area. Thus if we can correctly account for binary relationships we can produce an encoding whose length is close to that of the global minimal-length encoding. Luckily, accounting for binary relationships turns out to be relatively easy.

We have four types of cost to consider in computing the minimal-length encoding. The first is the part description cost, which is a constant number of bits for each hypothesized part descriptor. Thus there are no part-to-part interactions for this portion of the encoding cost. The second is the noise cost, the cost of encoding a pixel value relative to the part's surface. Here we have binary interactions, because the same pixel can be covered by more than one part. Similarly, in calculating the cost of errors of commission we must take binary relationships into account, because two parts can overlap the same pixel.

To account for these binary relationships we form what we call *area matrices*. We let $\mathbf{M}_a$ be a matrix whose i^{th} diagonal element is the number of pixels covered by the i^{th} part.

The $(i,j)^{th}, i \neq j$ elements of this matrix are minus one-half the number of pixels covered by *both* the i^{th} and j^{th} parts. A similar matrix, M_c is formed for the pixels which are errors of commission.

Let us describe a particular set of models by the vector $\vec{X}$, which has a one in the i^{th} slot if model i is a member of the set, and zero otherwise. To calculate the total cost of encoding image data by use of the set of part models $\vec{X}$ we evaluate

$$E(\vec{X}) = \kappa_0 \vec{X}\vec{X}^T + \kappa_1 \vec{X}\mathrm{M}_a\vec{X}^T + \kappa_2 \vec{X}\mathrm{M}_c\vec{X}^T + \kappa_3(N - \vec{X}\mathrm{M}_a\vec{X}^T) \tag{6}$$

where κ_0 is the cost of describing one part model, κ_1 is the cost of encoding a pixel to within $\pm 1/2\sigma$, κ_2 is the cost of encoding an error of commission, κ_3 is the cost of encoding an error of omission, and N is the number of non-zero pixels in the image.

By minimizing Equation(6) we can thus approximate the minimal-length encoding of the image data using our shape vocabulary.

Equation(6) is a quadratic form in $\vec{X}$; the minimum value of this equation can be found by gradient ascent as long as the combined matrix

$$\mathrm{M} = \kappa_0 \mathrm{I} + (\kappa_1 - \kappa_3)\mathrm{M}_a + \kappa_2 \mathrm{M}_c \tag{7}$$

is negative definite. This condition obtains whenever the part models do not overlap each other greatly, i.e., whenever our initial local encodings are reasonable.

The major difficulty in minimizing this equation is that the values of $\vec{X}$ must be either zero or one. In the following examples we have used a simple iterative, best-first search algorithm that is similar to the Newton-Raphson iterative solution technique. This minimization problem can also be converted to a linear integer programming problem, and global solution obtained by a method similar to the Simplex algorithm. We are currently investigating this linear approach.

By minimizing Equation (6) we obtain 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 step of optimization on all of the parameters of this final set of parts. 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 encoding cost; the algorithm then changes one of the part's parameters and ascertains whether the cost is improved. If improvement does indeed occur, the change is accepted.

3.6 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 [28], Brooks [29], Goad [30], Faugeras *et al* [31], or Grimson and Lozano-Perez [32] in their model-based 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 thus the required computation time can be significantly reduced. Because of the inherent parallelism of the technique (thousands of identical searches within each region) a full global search is expected to take only a few seconds per image on a large, parallel computer such as a Connection Machine.

4 PART RECOVERY RESULTS

The examples below all use range imagery: one synthetic example, one structured-light example, and three time-of-flight laser range finder examples. One characteristic of range data is that it is generally easy to obtain a rough "figure/ground" separation [23]; we have made use of that ability. Some simple preprocessing was used in the case of the laser rangefinder data, to remove mixed-range pixels and the inherent ambiguity-interval problems of those data [33].

4.1 Synthetic Data

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

Figure 7(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 7(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 moved about. Figure 7(c) shows the initial explanation of the image data, found by our local, parallel search technique applied to points sampled along the figure's 2-D skeleton, [13] followed by our procedure to select an optimal subset from among the regionally-best-fitting part models.

The most striking aspect of 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 7(d) shows the final recovered model, the result of gradient descent from Figure 7(c). Figure 7(e) shows "blow-up" views of the original model and of the recovered model.

[13] Originally the skeleton was found by hand simulation of a grassfire technique; later we confirmed that the points we used could have been found automatically.

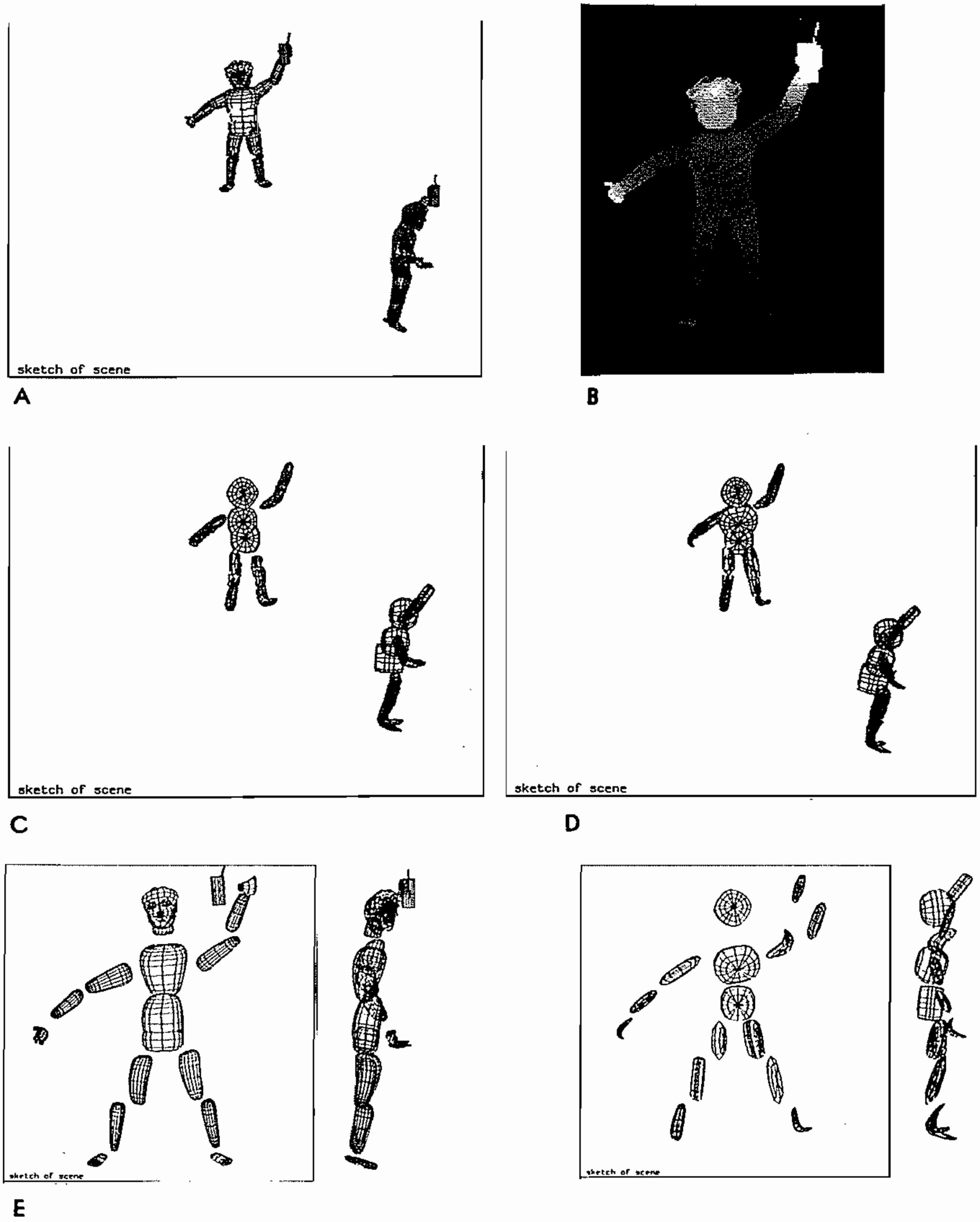

Figure 7: (a) A SuperSketch model, (b) a range image generated from this model, (c) initial explanation of the image data, (d) final recovered model, (e) "blow-up" views of the original model and of the recovered model. Note the similarity of part structure and part-by-part parameters.

Note the similarity of part structure and 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*, because stability in structuring the image data is the primary property required for reliable recognition. 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, although 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 — again, the upper part is occluded in this case by the "forearm." Although these assignments of part structure are not ideal they are entirely plausible segmentations when given only one view. It seems that such occasional merging of connected primitives may be unavoidable; thus in order to have reliable recognition we must have *both* possible descriptions — as one bent primitive and as two straight but connected primitives — in our stored model.

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 [9] pointed out the need for this type of "summarizing;" they proposed that for reliable recognition we must have a *multiscale* representation in our stored model. In this example we can see how having a multiscale representation, with descriptions for each distinct scale of parts structure, might be combined with this recovery procedure to successfully address some of the difficult problems associated with scale.

4.2 Industrial Castings

Figure 8(a) shows a range image of jumbled industrial castings produced by light striping. This image is not geometrically accurate; i.e., perpendicular surfaces do not appear perpendicular. The dynamic range of the image is about six bits. Figure 8(b) shows the initial explanation for the image data, found by parallel search at points along the figure's 2-D skeleton, followed by selection of the optimal subset. Again, it appears that the part structure structure has been recovered with sufficient accuracy to allow identification of the castings and recovery of their approximate location and orientation.

Figure 8(c) shows the result of gradient descent from Figure 8(b). Figure 8(d) shows a comparison of the original range data and a range image produced by the recovered model of Figure 8(c); the point of this comparison is that the simple (approximately 70 parameter) description that was finally recovered retains most of the data's original metric information. Such a good fit shows that we have achieved an accurate encoding of the data, and that our model vocabulary has descriptive adequacy for this data.

4.3 Outdoor Vision

The remaining examples make use of data from the ERIM Autonomous Land Vehicle time-of-flight laser range finder. This rangefinder collects a 256 by 64 pixel image in 0.4 seconds,

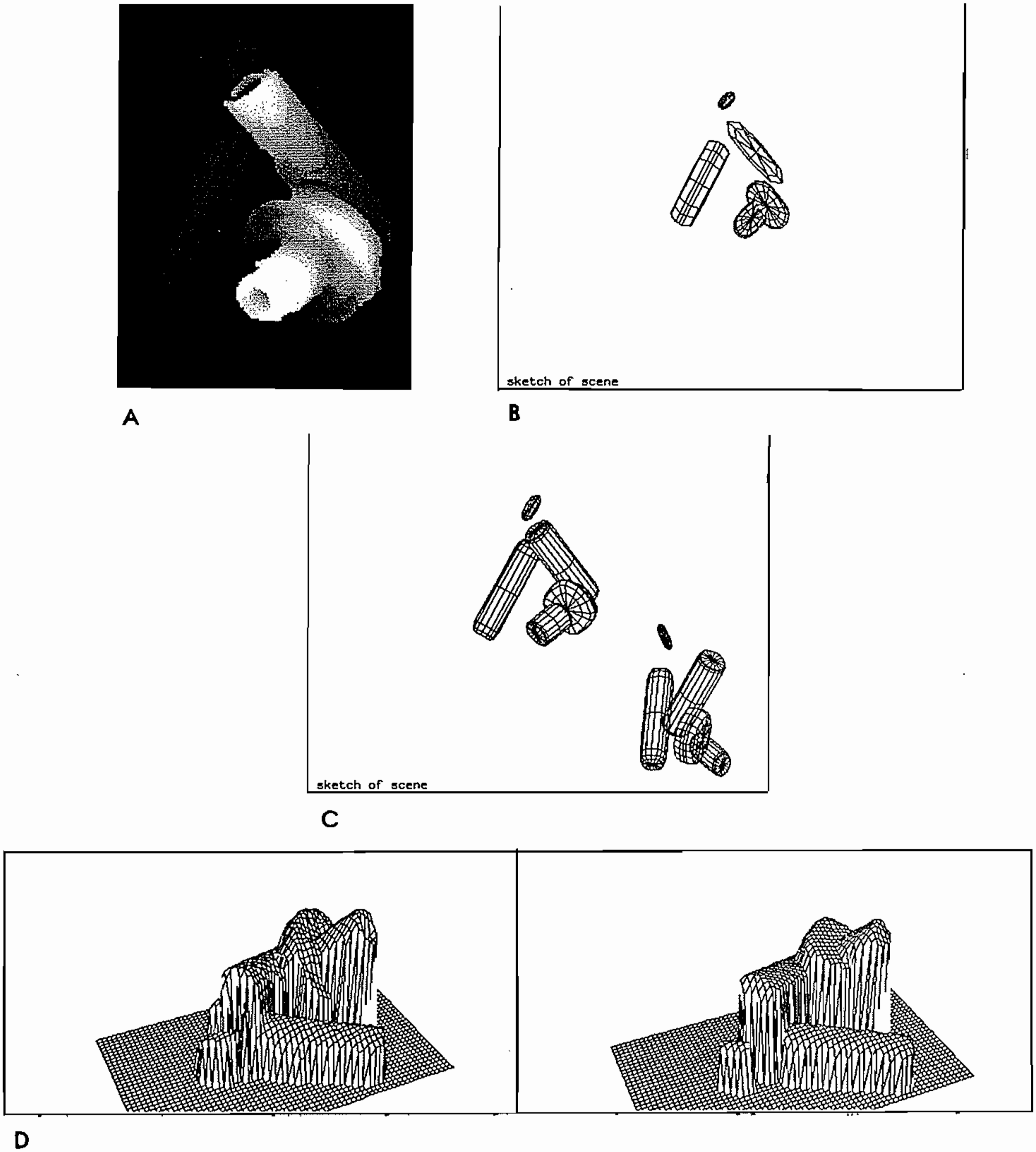

Figure 8: (a) A range image of jumbled industrial castings produced by light striping, (b) initial explanation of the image data, (c) the final recovered model, (d) comparison of the original range data and a range image produced by the final recovered model.

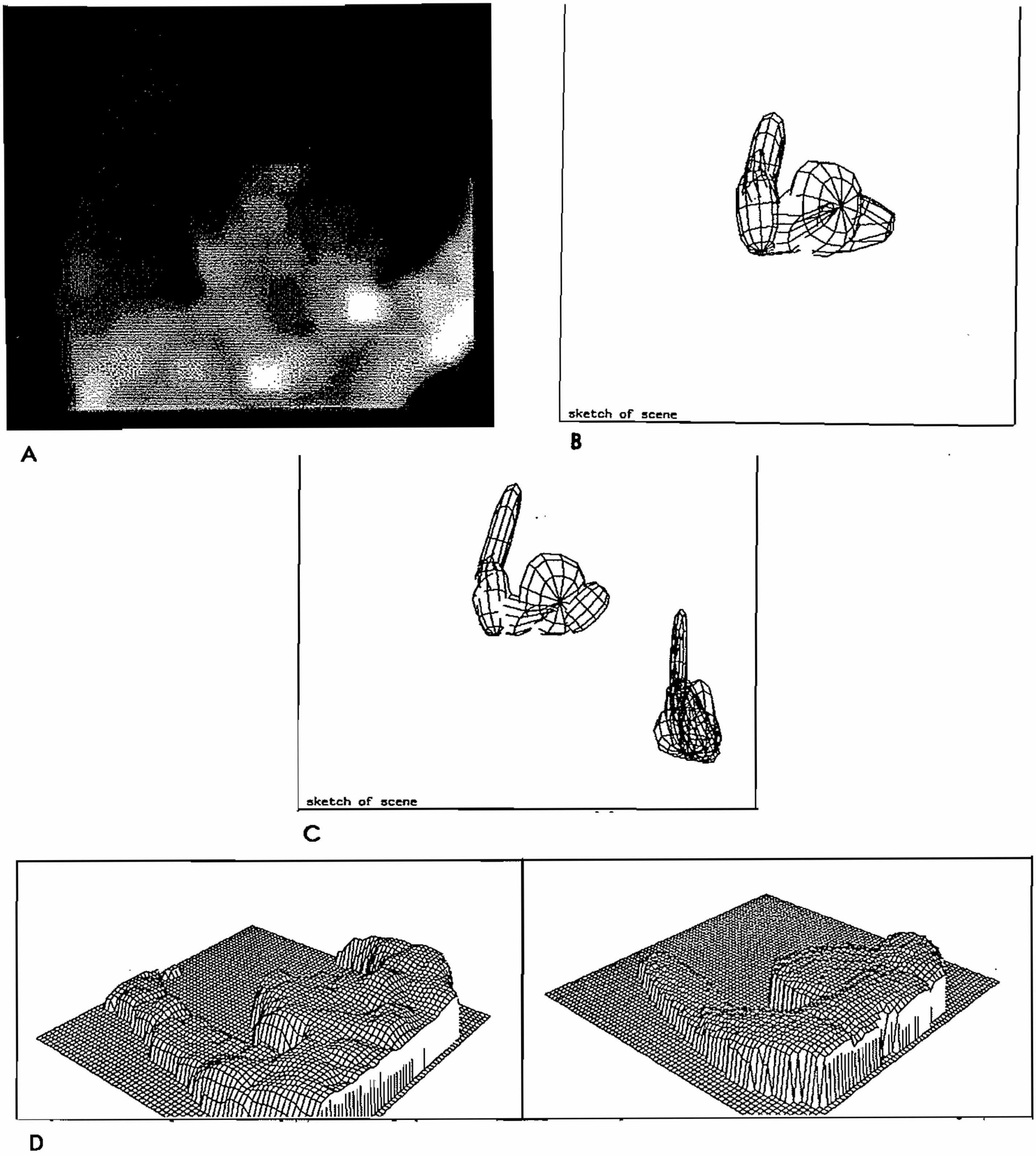

Figure 9: (a) A range image of the upper part of a person (for practical purposes this is merely a silhouette), (b) the initial explanation of the image data; note that the correct part structure of his left (upraised) arm, head, and right arm are clearly present, (c) the final recovered model, (d) a comparison of the original range data and a range image produced by the final recovered model.

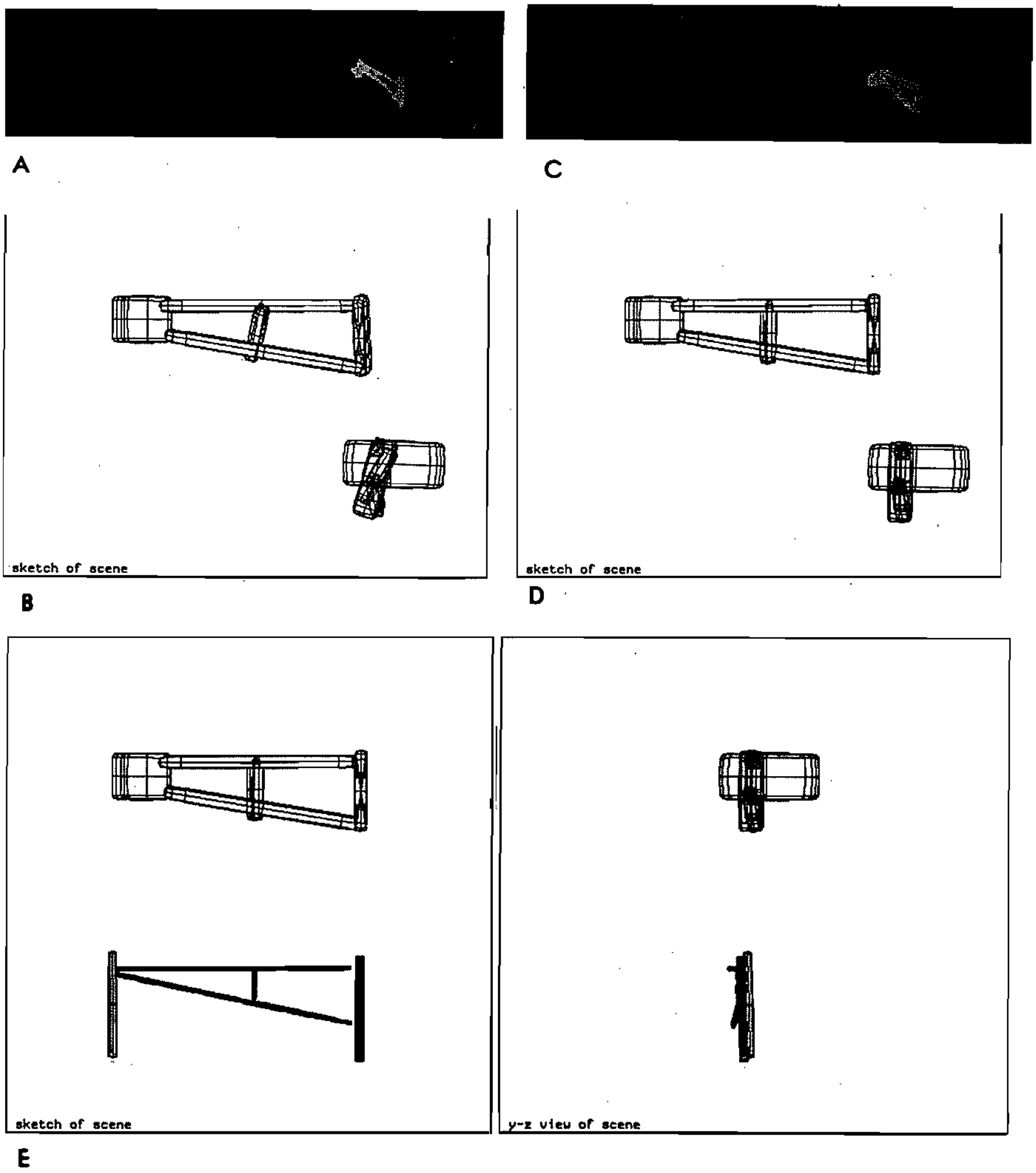

Figure 10: (a) A range image of a gate by the side of the road (our figure/ground procedure has, unintentionally, included a bush near the left-most gatepost as part of the figure), (b) the final recovered model, (c) a range image produced by the recovered model, (d) the result of "prettifying" (b) using the domain knowledge that nearly horizontal/vertical parts are likely to be horizontal/vertical, (e) a comparison of the recovered model of (d) with a SuperSketch model of the gate that was constructed by hand.

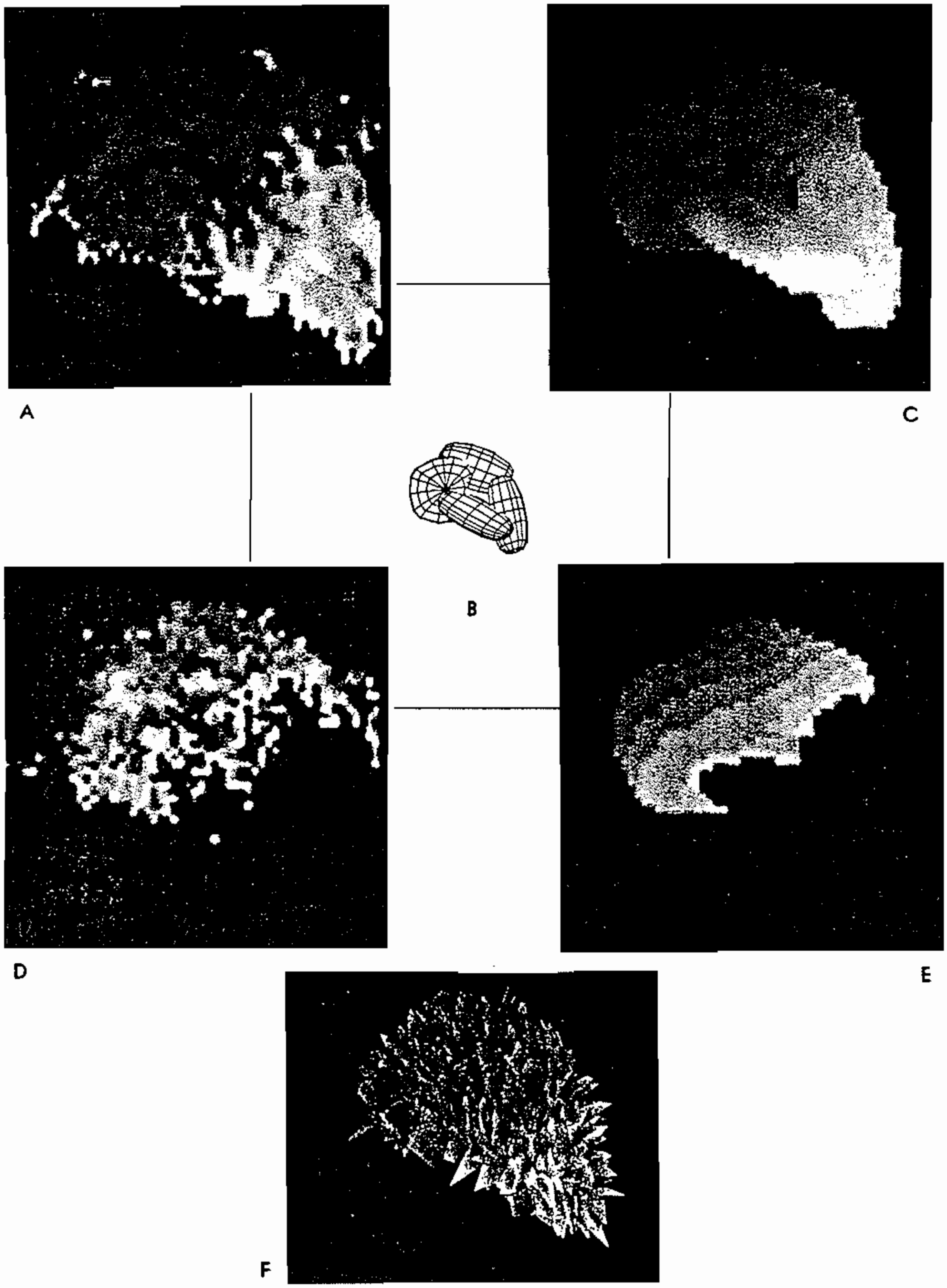

Figure 11: (a) A range image of a few roadside bushes, (b) final recovered model, (c) a range image produced by the recovered model, (d) the original range data, with points closer than the median distance removed, (e) a range image produced by the recovered model again with points closer than the median distance removed, showing the model's close match to the range data's internal structure, (f) a range image produced by adding a fractal surface model to the parts representation.

and has a useful range of about 128 feet and an advertised accuracy of about five percent. It is has an unusual imaging geometry that is similar to that of a very-wide-angle lens.

Figure 9(a) shows a range image of the upper part of a person, taken with this sensor. This example is interesting, especially in comparison to the synthetic data example above, in that the amount of depth information within the figure is negligible; this is for practical purposes merely a silhouette. Figure 9(b) shows the initial explanation of the image data; Figure 9(c) shows the result of gradient descent from Figure 9(b).

Perhaps the most important point of this example is that a reasonable 3-D part structure can be recovered even from what is essentially only silhouette data; the left (upraised) arm, head, and right arm are clearly present in the recovered description. Figure 9(d) shows a comparison of the original range data and a range image produced by the recovered model of Figure 9(c). This comparison shows that the simple (70 parameter) recovered description retains most of the data's original metric information, again demonstrating that we have achieved an accurate encoding of the image data and that our vocabulary has descriptive adequacy for this data.

Figure 10(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 unintentionally 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; this data thus provides a severe test of noise sensitivity.

Figure 10(b) shows the initial explanation of the image data. Again, a reasonable part-structure description is recovered, although (because the data include a bush as well as the gatepost) the left gatepost is recovered as a large "block" shape. Figure 10(c) shows a comparison of the original range data and a range image produced by the recovered model of Figure 10(b); again, the recovered description retains most of the data's original metric information, demonstrating that we have achieved an accurate encoding of the image data.

One of the advantages of having a high-level description like this parts language is that it provides good "hooks" for applying domain-specific knowledge. This is illustrated by Figure 10(d), which shows the result of "prettifying" Figure 10(b) using the domain knowledge that nearly horizontal/vertical parts are likely to be horizontal/vertical. Figure 10(e) shows a comparison of the recovered description of Figure 10(d) with a SuperSketch model of the gate that was constructed by hand. The thickening of the posts and bars in the recovered gate can be largely attributed to preprocessing intended to remove mixed-range pixels.

Figure 11(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. This data, therefore, allows us to examine some of the limits of our technique's descriptive adequacy, its ability to produce a reasonable encoding of even very noisy data, and its ability to produce stable segmentations. Figure 11(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 by 10 grid covering the image data.

Figure 11(c) shows a range image produced by the recovered description of Figure 9(b); the outlines of Figures 11(a) and (c) can be seen to be similar, thus showing that the recovered description (in this example a total of 56 parameters) retains most of the data's original metric information. Figure 11(d) shows a comparison of the original range data, with points closer than the median distance removed, and a range image produced by the recovered description of Figure 11(b), again with points closer than the median distance removed. This comparison shows that the *internal* structure of the recovered description 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 [11,34]. The result of adding a fractal surface model to the recovered part model is shown in Figure 11(e). An even better description of the differences between part model and image data could be obtained by using a particle model [35] of the bush's branches and leaves.

5 SUMMARY

We have used the approach of minimal-length encoding to address the problem of recovering volumetric object descriptions from range data. This technique provides us with the "Occam's Razor" simplest explanation of the data, an explanation that can be shown to be the maximum a posteriori (MAP) estimate of the scene's structure in terms of our volumetric shape vocabulary.

When using this approach it is critical to use a shape vocabulary that "naturally" describes the scene's true 3-D structure. We have used various avenues of psychological evidence to settle on a representation that closely mimics people's notions about object's intrinsic part structure [4-6,8,12-14,19]. By use of the SuperSketch modeling system we have been able to show that this volumetric shape representation is reasonably general purpose despite having only a small number of parameters. Further, we note the fact that we could in each example obtain an accurate fit to real range data lends support for the shape vocabulary's descriptive adequacy.

In order to obtain a solution to the minimal-length encoding problem, we developed a two-stage optimization procedure that first addressed the local encoding problem, and then the global encoding problem.

The first stage capitalized on the fact that our shape models have only a small number of parameters, allowing us to use a massively parallel search to determine points in the model's parameter space that potentially provide the optimum local encoding of the data. This technique has the advantage that it can reliably obtain a *global* solution for certain difficult non-linear optimization problems, such as the problem addressed here, by use of a small number of massively parallel operations. Further, this parallel search technique has considerable plausibility as a model of biological information processing.

The second stage of our encoding procedure chooses among these local encodings to obtain an approximation to the global minimal-length encoding. This problem may be formulated as a quadratic integer programming problem, and gradient descent techniques used to search for an optimum solution. Alternatively, the second stage may be converted to a linear integer programming problem and the exact solution obtained, an approach we are currently investigating.

From our experiments we believe that this parallel-search minimal-length encoding technique is robust. The examples presented here, for instance, demonstrate stability with respect to both noise and scale. Thus our approach seems to be able to provide a good encoding of the data's metric properties by use of a small number of massively parallel operations followed by a straightforward quadratic optimization.

More important than obtaining a good encoding of the data's metric properties, of course, is obtaining an accurate segmentation of the object into it's *part structure*, for that is what we must use for indexing into our memory of stored models. Whether a model that accurately accounts for the metric properties also accounts for the part structure depends upon whether our shape vocabulary actually captures a robust aspect of the object's true 3-D structure, and whether that structure is conserved under projection.

It is our opinion that the part structure descriptions we have been able to recover are in fact adequate to support recognition. For instance, Figure 7(e) makes it seem quite likely that we could to compare our recovered part description of the man image (Figure 7(d)) with a stored description of the object category "a man" (Figure 7(a)) and determine that the recovered description is in fact an image of a man. Further, it seems likely that we could determine that the recovered description of Figure 9(c) is at least consistent with our man model. Whether or not our current system is in fact sufficiently robust to generally support such recognition is, of course, open to experimental verification, and consequently is the principal direction of our future research.

REFERENCES

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

[2] 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.

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

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

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

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

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

[8] 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.

[9] Marr, D. and Nishihara, K., (1978) Representation and recognition of the spatial organization of three-dimensional shapes, *Proceedings of the Royal Society - London B*, 200:269-94

[10] Terzopolis, D., (1987) Regularization of inverse visual problems involving discontinuities, *IEEE Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 413-424.

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

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

[13] 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.

[14] 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

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

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

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

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

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

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

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

[22] Bajcsy, R. and Solina, F., (1987) Three-dimensional Object Representation Revisited, *First International Conf. on Computer Vision, '87,* June 8-11, London, England.

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

[24] Geman, S., and Geman, D., (1984) Stochastic relaxation, gibbes distributions, and the Bayesian restoration of images, *IEEE Trans. on Pattern Analysis and Machine Vision,* Vol. 6, No. 6, pp. 721-741.

[25] Witkin, A., Terzopolis, D., and Kass, M., (1986) Signal matching through scale space, *Proc. National Conf. on Artificial Intelligence '86,* pp. 714- 719, Philadelphia, PA.

[26] Solina, F., and Bajcsy, R., (1987) Range image interpretation of mail pieces with superquadrics, Proc. Am. Asso. Artificial Intelligence '87, pp. 733-737, Seattle WA.

[27] 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.

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

[29] 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.

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

[31] 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

[32] 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.

[33] 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

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

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

Appendix

We can show that, ignoring the (x, y, z) position parameters, that there are less than 2^8 local maxima when searching for the optimum fit between our SuperSketch models and range image data. We accomplish this by means of the following polynomial description of the set of SuperSketch models.

We start by defining a three-sphere of radius $r = a_0$:

$$x_0^2 + y_0^2 + z_0^2 = r^2 \tag{8}$$

We then define translation

$$x_1 = (a_1 + x_0) \qquad y_1 = (a_2 + y_0) \qquad z_1 = (a_3 + z_0) \tag{9}$$

and three-space rotation

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = R(a_4, a_5, a_6) \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \tag{10}$$

where $R(a_4, a_5, a_6)$ is a rotation matrix corresponding to the three Euler angles a_4,a_5, and a_6.

We then introduce a series of transformations that correspond to the deformations allowed in SuperSketch. First we define stretching,

$$x_3 = a_7 x_2 \qquad y_3 = a_8 y_2 \qquad z_3 = a_9 z_2 \tag{11}$$

then bending (here we will use only bending around the z-axis),

$$z_4 = a_{10} x_3^2 + a_{11} y_3^2 + z_3 \tag{12}$$

and finally tapering,

$$r = 1 + a_{12} z_4. \tag{13}$$

Thus the equation

$$x_3^2 + y_3^2 + z_4^2 = r^2 \tag{14}$$

defines the set of ellipsoidal SuperSketch models, where the variables a_i correspond to the parameters of our SuperSketch models. From this equation we can derive a single-valued function

$$z_4 = f(x_3, y_3, a_i) \tag{15}$$

that describes the visible face of a SuperSketch models with parameters $< a_i >$.

We now consider the minimization problem

$$\epsilon = \min_{a_i} \sum_j \| z_j - f(x_j, y_j, a_i) \| \tag{16}$$

that is, find the set of a_i that best describe a set of range data $< x_j, y_j, z_j >$. This minimization equation is a fourth-order polynomial in the first eight [14] a_i, and a second-order polynomial in the remaining a_i.

Simple calculation shows that, ignoring the positional parameters, there are *at most* $3^5 = 243$ local maxima in this optimization. Because not all cross terms of the a_i are typically present, however, there are usually fewer maxima than this simple calculation would suggest. We also observe that position, orientation, length, and breadth may need to be sampled fairly densely, as they potentially have three maxima apiece, whereas the remaining parameters are quadratic and thus can be established by gradient ascent.

We can extend this argument to squarish SuperSketch models by modifying the original equation to read:

$$x_3^4 + y_3^4 + z_4^4 = r^4 \tag{17}$$

The use of the fourth-order power changes the surface from a uniformly curving one to a surface with "corners." Although the order of the equation is now changed, there are no additional *distinct* roots, and so the number of local maxima is unchanged.

Perhaps the main problem with implementing an optimization procedure based on this analysis is that we don't know the position of the maxima or their characteristics, so that we don't know how to sample the parameter space. One approach is to evenly tile the parameter space with starting points separated by an amount smaller than the diameter of the convex region surrounding the global optimum. Although such a uniform grid will inevitably employ many more starting points than are actually necessary, it is an approach that requires relatively little knowledge about the global behavior of the optimization function; one only needs to know about the region immediately adjacent to the global optimum.

This approach still requires an estimate of the diameter of the convex region surrounding the global optimum. One method we have developed to empirically determine the correct sampling density we call the *expanding grid method.* We start with a massive *under*estimate of the diameter, and build a grid of that density across the entire parameter space.

We then test the sufficiency of this estimate by checking to see that gradient ascent from each sampled point can reach at least one neighboring point in the parameter space. If our test succeeds, then we increase the interpoint distance, continuing until we reach a point where our test fails. Assuming that there are no maxima whose diameter is less than our original (under)estimate, then this procedure gives us a useful estimate of the diameter of the convex regions surrounding the global maximum.

The final step is to determine the maximum ratio of goodness-of-fit values that occurs within *every* global maximum. In our experiments with fitting SuperSketch models to range data, there was always at least a factor of three between the optimal fit and other points still within the convex region surrounding the global maximum (discounting certain intrinsic parameter ambiguities). This ratio was then used to determine which points cannot possibly be within the convex region surrounding the global maximum.

[14] Although the Euler angles do not appear as linear elements of the rotation matrix, it is only necessary to obtain three independent elements of the rotation matrix in order to solve for them; these independent elements are therefore the relevant rotation parameters, and they are linear elements of the rotation matrix.