PART STRUCTURE FOR 3-D SKETCHING

Alex P. Pentland

Artificial Intelligence Center, SRI International
333 Ravenswood Ave, Menlo Park, California 94025
and
Center for the Study of Language and Information
Stanford University, Stanford California 94038

ABSTRACT

Natural, efficient communication depends upon shared representations. Current 3-D graphics systems, however, use representations that are quite distant from that which people use. The result is that construction of 3-D models is much like programming: meticulous translation from the persons' internal representation to the machines' representation. We argue that a constructive solid geometry representation that allows stereotyped deformations and statistical specification closely parallels peoples' internal representation. Such correspondence allows fast, "natural" 3-D modeling; this is especially important in the initial stages the design process where a "sketching" capability is more important than the ability for precise control of details. We describe and evaluate an interactive system that uses such a representation. The system demands real-time interaction; to support this on 68020-class machines we develop a linear-time hidden line algorithm, so that the hidden-line calculation requires only slightly more time than is needed to draw the lines.

1 Sketching versus Detailing

The distinction between *sketching* and *detailing* is important in understanding how people create a 3-D model. For instance, engineers typically sketch a new part using paper and pencil, and then give the sketch to a draftsman who uses a CAD system to complete the detailed specification of the model. Similarly, animators sketch out scenes and actions before drawing careful renditions of the sequence. The reason that people standardly divide the design process into two stages — each employing its' own media — is that there are two conflicting sets of requirements: the initial design of a 3-D model (i.e., 3-D sketching) demands the ability for quick, general-purpose, and natural interaction, while the final drafting or rendering stage demands the ability for detailed, precise control.

Most current 3-D graphics systems have the wrong "control knobs" for the initial, sketching phase of the design process; that is, the things you would like to do when "roughing in" a 3-D model aren't usually easy to do. This makes things difficult; you have to approach the task of modelling a shape in a planned, methodical manner, much as a programmer approaches the problem of constructing a program[1] Because you have to carefully plan your interaction with the machine, both engineers and graphic artists still sketch shapes on paper before attempting to use a 3-D modeling system.

The use of paper for sketches and computers for final models is bad for exactly the same reasons that the use of paper for final models is bad: lack of flexibility, unneeded duplication of effort, no library of previous drawings, and so forth. In an attempt to address these problems we set out to develop a 3-D modeling language, user interface, and rendering system that is sufficiently "natural" and interactive that people would choose to sketch shapes on the *computer* rather than sketching them out on *paper*.

The idea, then, was to develop a tool that allows the user to *very* quickly build or modify a 3-D model; to replace the pencil and paper. A user would directly *sketch* 3-D form on the computer, playing with the shape until it looks right, rather than approaching the modeling task as one of entering a carefully predefined model into the computer. An engineer would quickly "sketch" a new part directly on the computer, playing with it until it satisfied him. An animator would "sketch" a scene and, Claymation-like, interactively modify the scene so as to step through key points in an action sequence. In both cases, once we are satisfied with this "sketch model," we can then invest the time to carefully fill out the models' details using a system that is specialized for that particular task.

We want, therefore, a tool that is not specialized to any one application domain but, like pencil and paper, is equally applicable to any 3-D modeling task. And further, like pencil and paper, we want this modeling tool to be generally *available*: i.e., cheap enough to sit one on everyones' desk, so that they will actually use it.

1.1 The Design of a Graphics System

We have implemented our solution to these problems in a system called SuperSketch (named for "sketching" and "superquadrics"), which provides an environment for interactively sketching and rendering 3-D models. The specific major design criteria for SuperSketch were:

(1) Representation: The system must have a communication metaphor (language) that closely matches the way people naively think about and discuss shape, to promote easy, natural communication between the user and the machine.

(2) Interaction: The system must have an interaction interface that allows users to attain a level of "effortless" interactive control similar to that of an engineer or artist sketching in pencil.

(3) Efficiency and Accessability: If it is to be truely useful, the system must be efficient enough to allow "real-time" line drawings and rapid full color renderings on a computer inexpensive enough to sit on everyones' desk; e.g., a Motorola 68020-class machine without additional hardware.

In the following sections of this paper we will discuss how we have sought to meet each of these design criteria.

2 Representation

The process of constructing and animating a 3-D model is a process of communication between the machine and the human operator. Because communication depends upon having a shared representation of the situation, the development of natural, "effortless" methods for constructing and animating 3-D shapes depends upon having a representation that is isomorphic to that which people use. When the representation used by the machine doesn't match the way the human operator thinks of

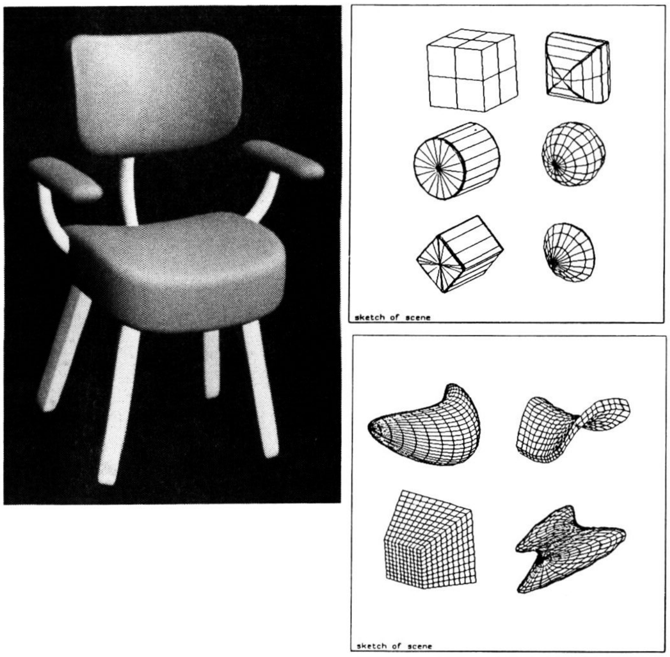

Figure 1. (a) A chair; naive subjects typically describe this as being formed from Boolean combinations of appropriately deformed modeling primitives, (b) a sampling of the basic forms allowed, (c) deformations of these forms.

the process, we get what I call the "Etch-A-Sketch problem[2] :" the system has the wrong control knobs

2.1 Man-machine interaction: building 3-D models

As an illustration of why the way you represent a scene is important, imagine that you were looking at a chair such as is shown in Figure 1(a), and trying to figure out how to build a 3-D model of it. When people verbally describe the shape of this chair, they typically [1,2] say things like

> "Well, the back of a chair is a sort of squarish, thin thing that has been bent slightly. The bottom of the chair is the same but thicker, and rotated 90°. The legs are long rectangular things stuck into the bottom of the chair, and ..."

People describe shapes in terms of combining "parts" to form prototypes, and in terms of certain standard deformations of those parts and prototypes. If the computer understood such descriptions, then you could enter the above description of a chair directly. You could construct a 3-D model almost as easily as you could produce a verbal description.

Typically, however, the representation the computer uses is more like splines or polygons, so to enter the model you must adjust spline control points or enter polygons vertices to obtain a shape that matches your mental image of the desired form. Unfortunately, people do not "see" or (normally) think of objects in terms of polygons or splines. Thus the user is forced to carefully (and laboriously) translate between his mental concept of the shape and the computers' representation — to "program" in the base language that the computer uses.

Thus we can liken building a 3-D model on most current day 3-D graphics systems to programming a computer in machine language: you can do anything, but it is often quite laborious. Nor will an elaborate human interface help much: such an interface is like providing the programmer with an assembly language and stepping debugger. Such tools are much better than machine language, but as long as the basic representation is unnatural for the user they still fall short of providing the advantages of a high level language.

Thus it seems that if we could discover a concrete, mathematical version of the "parts" that people use to think about 3-D shape, we could construct a graphics system that wouldn't require the user to be a programmer: it wouldn't require him to translate from the way *he* thinks of the problem to the way the *computer* represents the problem.

2.2 Animation

Similar problems arise when we turn from the problem of building 3-D models to the problem of animating them. Polygonal representations, for instance, are too fine grain for ease of manipulation; often the path of each polygon must be separately controlled to produce natural motion. Similarly, spline representations have the problem that non-rigid motions require a very difficult-to-compute interpolation of the spline parameters.

These difficulties arise because the grain size of the representations doesn't match grain size of the problem. Points in the world are not, typically, independent of each other — as they appear in fine-grained polygonal representations — they often move in concert, rigidly or elasticly. Larger grain representations such as splines or Constructive Solid Geometery (CSG) systems have the opposite problem, as they assume the relationship between points to be fixed: animators, unfortunately, often want objects to move elasticly, and to stretch or compress.

For animation we need to have a representation that matches the grain size of the problem. The disciplines of mechanics, dynamics and kinematics provide a suggestion about how to represent objects for animation, for they represent objects as fixed, solid bodies that undergo translation, rotation and elastic or inelastic deformation.

To model a blade of grass bending in the wind, for example, we would probably first take our polygon or spline description and find a simple mathematical model that was "similar", e.g., a rigid rod. We would then compute the deformation caused by the wind pushing evenly along the length of the rod, and then finally map that deformation back to the polygon or spline representation of the actual shape. It is obvious that things would be simpler if our original representation for the blade of grass were the same one we used for computing the parameters of the bending motion; e.g., a single mathematical object, like the rod, that could then be deformed and rendered directly.

As a more complicated example, consider the modeling of vibrational modes in the animation of biological forms. Muscles, joints and flesh are elastic, and so realistic biological motion must include bouncing and elastic deformation as well as translation and rotation; perhaps the best illustration of this is found in Walt Disneys' movies, e.g., the dancing dwarfs in "Snow White and the Seven Dwarfs."

When analyzing the vibrational modes of objects, the standard proceedure is to break complex shapes into the union of simple convex shapes whose compression, extension and bending may be separately considered. Thus if we represent our shapes as unions of convex forms with later deformations — similar to the "parts" that people naturally use to describe shape — we will be more easily able to describe, compute, and constrain the parameters of motion because they will be relatively simple functions of the description. That is, a part-by-part description will provide the right "control knobs" for computing the parameters of motion.

In summary, then, the fact that a "part" description is the basis for both peoples' naive notions of form and for mechanics/dynamics/kinematics makes it seem likely that we can develop a descriptive vocabulary that will allow us to accurately model the world in terms of *parts*: a parameterized set of volumetric primitives that, in relatively simple combination, can be used to form rough-and-ready models of the objects in our world and how they behave. If we can develop such part-like modelling primitives then not only will animation become easier, but the problem of building 3-D models will become easier because people seem to think about shape in terms of such part descriptions. The first question to be answered, therefore, is what is the notion of "a part" that people use?

2.3 People: Parts and Collective Abstractions

A considerable amount is known about how people conceptualize 3-D shape. For instance, we have found that the chair example above is a general phenominon — i.e., people describe form in terms of combinations of component parts, which in turn are described as modifications of standard prototypes. This sort of structuring of imagery was first explored by the classical Gestalt school of perceptual psychology [3,4], and today is the subject matter of a lively school of investigating human perception [5,6,7]. Indeed, such a part-based, prototype-and-modification descriptive system seems to be common to all human spatial reasoning; the classic work by Rosch [8], for instance, supports this view: she showed that even primitive New Guinea tribesmen (who appear to have no concept of regular geometric shapes) form the geometric prototypes in much the same manner as people from other cultures, and describe novel shapes in terms differences from these prototypes.

Nor is this purely a cognitive phenominon. When images are stabilized on the retina, for instance, they seem to disappear because low-level mechanisms in the human visual system suppress anything that doesn't move. [This is why you don't see the veins in your retina.] What is interesting is that this disappearance doesn't occur uniformly, but rather affects things in chunks: whole "parts" of objects fade and return, rather than line segments, random patches, or whole objects [9,10]

The central consensus of this research is that people see part boundaries as occuring at places of extremal curvature or at inflections[3] ; this leads to a characterization of 3-D parts as being Boolean combinations (specifically or's and not's) of convex "blobs" [11,12]. When there are specialized cues that indicate that two portions of a figure share a common history — e.g., pronouced axes of symmetry, parallelism, etc. — the human visual system groups these portions together into a single "part" [6,13,14]. Thus we must allow certain stereotyped deformations of our convex blobs to still be considered as a single "part." But which deformations?

We have found that in verbal descriptions of unfamiliar imagery (electron microscope images) people commonly employ a limited set of deformations: bending, tapering, and twisting [1,2,14]. We can also address the question of which deformations are allowable by examining the range of image cues that support the perception of a "deformed part." When we do this we find that the most important grouping cues — symmetry and parallelism - allow reliable inference of bending and tapering, and perhaps of twisting in the case of square-edged or ruled forms [6,13]. Thus we will adopt bending, tapering and twisting as our sole allowable deformations.

Complex natural surfaces. Things seem to happen somewhat differently, however, for complex natural forms such as clouds or mountains, perhaps because such natural shapes simply have too much detail to completely remember, and the details are too variable across instances of the same type of object. Experiments in human memory [15] suggest that for complex surfaces, e.g., a crumpled newpaper, people seem to abstract out a few properties such as "crumpledness" and a few major features of the shape such as the general outline. The rest of the structure is ignored; it is unimportant, *random*.

The fractal-like stochastic representations recently developed in computer graphics mimic this sort of abstraction of qualitative properties like "crumpledness" by letting us qualitatively describe the morass of details by means of a statistical process.

Interestingly, we have found that the parameters of these stochastic processes have a surprising amount of psychological reality. We have shown [16,17], for instance, showing that peoples' perception of "roughness" versus "smoothness" varies as a linear function of the surface's fractal scaling parameter ["fractal dimension"]. This result indicates that representations that incorporate such stochastic models are a start towards duplicating the sort of physically meaningful abstraction of shape that people accomplish.

2.4 A Representational System

The above considerations lead us to the following representational system, a system that we have found competent to accurately describe an extensive variety of natural forms (e.g., people, mountains, clouds, trees), as well as man-made forms, in a succinct and natural manner. The idea behind this representational system is to provide a vocabulary of models and operations that will allow us to model our world as the relatively simple composition of component "parts," retreating to statistical description when the complexity of the scene becomes too large for convienient manipulation.

The most primitive notion in this represention 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 basic modeling element we use a parameterized family of shapes known as a *superquadrics* [18,19], which are described (adopting the notation $\cos\eta = C_\eta$, $\sin\omega = S_\omega$) by the following equation:

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

where $\chi(\eta,\omega)$ 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 1(b). Superquadrics are, therefore, a superset of the modeling primitives commonly used in CSG systems.

These basic "lumps of clay" (with various symmetries and profiles) are used as prototypes that are then deformed by stretching, bending, twisting or tapering, and then combined using Boolean operations to form new, complex prototypes that may, recursively, again be subjected to deformation and Boolean combination. As an example, the chair in Figure 1(a) was constructed in much the manner that we have found people describe this shape: the back and seats are rounded-edge superquadric "cubes" that are flattened along one axis, and then bent somewhat to accommodate the rounded human form, etc.

The mathematical basis for this portion of the descriptive language was originally developed by Barr [20], although he did not envision it as the basis of a general purpose modeling language. Nonetheless, his work has let us develop a vocabulary of form that closely mimics human notions of part structure and is considerably more powerful than traditional CSG representations.

To illustrate the flexiblity of this representation, consider the range of basic superquadric shapes, as shown in Figure 1(b). Already this is a superset of traditional modeling primitives, as it includes rounded shapes as well as traditional Platonic solids. By allowing the deformations that people employ in verbal descriptions — stretching, bending, tapering and twisting —we greatly expand the range of primitives allowed, as shown in Figure 1(c).

Still, the most powerful notion in this language is that of allowing Boolean combination of the primitives. This intuitively attractive CSG approach — building specific object descriptions by applying the logical set operations "or" and "not" to component *parts* — introduces a language-like generative power that allows the creation of a tremendous variety of form, as is illustrated by the figures in this paper.

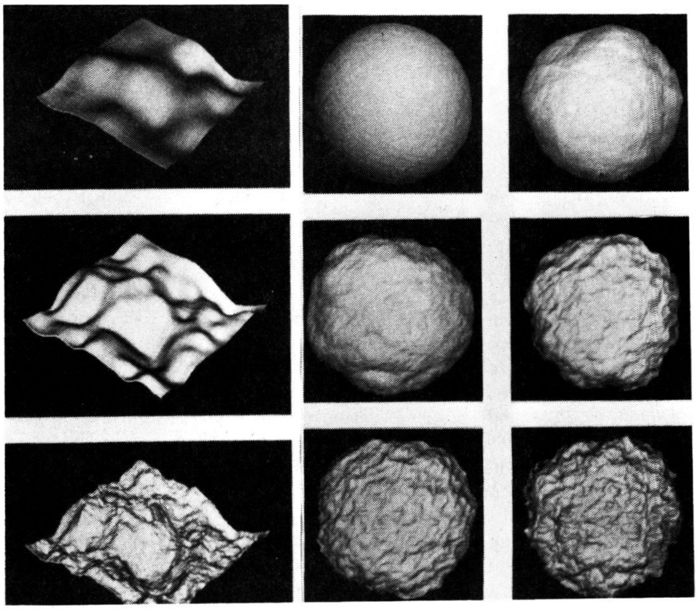

Figure 2. (a) - (c) show the construction of a fractal shape by successive addition of smaller and smaller features with number of features and amplitudes described by the ratio $1/r$. (d) Spherical shapes

2.5 Complex inanimate forms

To show how we may integrate the "part" representation discussed above with the textural abstractions needed to describe complex forms, let us first investigate a model of 3-D texture widely used in the graphics community: fractal Brownian functions. We randomly place n^2 large bumps on a plane (where n is a constant chosen so that the bumps adequately fill out the plane), giving the bumps a Gaussian distribution of altitude (with variance σ^2), as seen in Figure 2(a). We then add to that $4n^2$ bumps of half the size, and altitude variance $\sigma^2 r^2$, as shown in Figure 2(b). We continue with $16n^2$ bumps of one quarter the size, and altitude $\sigma^2 r^4$, then $64n^2$ bumps one eighth size, and altitude $\sigma^2 r^6$ and so forth. The final result, shown in Figure 2(c) is a true Brownian fractal shape. The validity of this construction does not depend on the particular shape of the superquadric primitives employed; the only constraint is that the sum must fill out the Fourier domain.

Different shaped lumps will, however, give different appearance or texture to the resulting fractal surface; this construction, therefore, lets us generalize the standard fractal constructive techniques to produce surfaces with varying lacunarity, etc. One particularly efficient way to produce such shapes is by convolution of appropriately scaled kernels over arrays filled with random noise[4].

When the placement and size of these superquadric lumps is random, we obtain the classical Brownian fractal surface that has been the subject of much previous research. When the larger components of this sum are matched to a particular object, however, we obtain a description of that object that is exact to the level of detail encompassed by the specified components.

This makes it possible to specify a global shape while retaining a qualitative, statistical description at smaller scales: to describe a complex natural form such as a cloud or mountain, we specify the "lumps" down to the desired level of detail by fixing the larger elements of this sum, and then we specify only the fractal statistics of the smaller lumps thus fixing the qualitative appearance of the surface. Figure 2(d) illustrates an example of such description. The overall shape is that of a sphere; to this specified large-scale shape, smaller lumps were added randomly. The smaller lumps were added with six different choices of r (i.e., six different choices of fractal statistics) resulting in six qualitatively different surfaces — each with the same basic spherical shape.

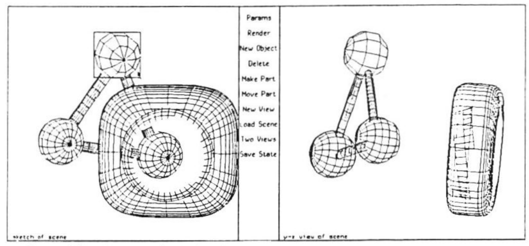

Figure 3. The SuperSketch viewports. The left viewport is an interactive view of the scene with hidden lines removed (the linear-time hidden surface algorithm is described in the following section). The right viewport is more like a wireframe model, so that objects are not lost to the users' view.

The ability to fix particular "lumps" within a given shape provides an elegant way to pass from a qualitative model of a surface to a quantitative one — or *vice versa*. We can refine a general model of the class "a mountain" to produce a model of a *particular* mountain by fixing the position and size of the largest lumps used to build the surface, while still leaving smaller details only statistically specified. Or we can take a very specific model of a shape, discard the smaller constituent lumps after calculating their statistics, and obtain a model that is less detailed than the original but which is still appears *qualitatively* correct.

3 Interaction

The first design criterion of our system is a representation (metaphor) that is natural to the human user. The previous section described the metaphor used in this system: that of building objects from clay, a descriptive strategy people often spontaneously use and which they find natural, using our superquadric-based analogy to the human perceptual notion of "parts." Thus the system presents the user with "lumps" of pliable material (like clay) that may then be formed by changing the parameters of the part-like primitives (e.g., modifying the squareness-roundness, length, amount of bending, etc.), and finally combined with other parts of the scene using boolean operations (e.g., "or" and "not").

The second design criterion of our system is that it have a user interface that allows users to attain a level of "effortless" interactive control similar to that of an engineer or artist sketching in pencil. To provide accurate, complete real-time interactive feedback of the state of the 3-D model under construction, we decided to employ two engineering-style orthographic views (x-y and y-z) of line drawing sketches of the scene. This is shown in Figure 3. All hidden lines are removed in the x-y view (labeled "sketch of scene"), but in the "y-z view" only external facing surfaces are rendered, i.e., objects are seen as "transparent" wireframes, with only back-facing or intersecting portions of the wireframe removed. This partial hidden-surface presentation prevents objects from being lost to the users' view. Objects can be moved, deformed, etc., and redisplayed using a two-hundred triangle line-drawing approximation to the underlying analytical form in about one-eighth of a second, thus providing the perception of smooth, "real-time" motion and deformation.

4 Efficiency and Accessability

Central to the user's impression of interactivity and "naturalness" is the real-time display of the current state of the 3-D model. Unfortunately, this requirement is in direct conflict with the criterion that our system run on Motorola 68020-class machines.

Polygon-based algorithms are fundamentally order $n \log n$ in the number of polygons, and z-buffer techniques, although linear in the number of polygons, are also linear in the number of pixels. Further, as we require the ability to perform Boolean combinations of our part primitives, we must also add in time for conversion to a standard polygon representation, which is typically order n^2. Thus achieving real-time display on these machines seems impossible with current algorithms, because their fundamental computational complexity.

We have therefore developed a hidden line algorithm that is linear in the number of polygons being modified. It is, as far as we have been able to determine, the only example of an incremental, linear-time algorithm other than z-buffer algorithms[6] . This algorithm may be viewed as an analytical version of ray casting [22].

5 Human Interaction Performace

We have set out to build a system that permits a user to quickly sketch a very wide range of form. How well have we really done? There are two ways to answer this question: One, have we developed a representation/metaphor that supports *natural* man-machine interaction?, and two, have constructed a system that permits quick, responsive modeling of form?. Although we have not yet done the sort of careful psychophysical testing that motivated our development of the representation, we can give a subjective evaluation and a few quantiative benchmarks; these are reported below.

A natural vocabulary?. We have found that, as a rule, when we try to model a particular 3-D form using this system we naturally tend to describe the shape in a manner that corresponds to the organization our perceptual apparatus imposes upon the image, even to making the distinctions standardly made in English. That is, the components of the description match one-to-one with our naive perceptual notion of the "parts" in the figure.

For instance, Figure 4 shows how the face is formed from the Boolean sum of several different primitives. The basic form for the head is a slightly tapered ellipsoid. To this basic form is added a somewhat cubical nose, bent pancake-like primitives for ears, bent thin ellipsoids for lips, and almond-shaped eyes, as is shown in Figure 4(a). Figure 4(b) show the addition of rounded cheeks and a slightly pointed chin (is this Yoda from Star Wars?), and finally Figure 4(c) shows the addition of a squarish forehead and slightly fractalized hair.

The smoothly shaded result is shown in Figure 4(d) — it is a reasonably accurate human head, composed of only 19 primitives, specified by slightly less than 130 bytes of information. The two scenes shown in Figure 5 are described in a similarly concise, natural fashion. Figure 5(a) contains only 56 primitives, or about 500 parameters/bytes of information. Figure 5(b) contains only 100 primitives (about 1000 parameters/bytes of information) despite the considerable detailing in the faces (see Figure 4). One should remember that this representation is *not* in any way tailored for describing the human form: it is a general-purpose vocabulary.

The extreme brevity of these descriptions is evidence of their "naturalness." We also note that this brevity makes many otherwise difficult tasks relatively simple, e.g., even NP-complete problems can be easily solved when the size of the problem is small enough. For instance, in animation one would like to be able to specify constraints like "x does not intersect y," "x attached to y," or even "x supports y." When even complex scenes can be described by relatively few "parts" the problem of satisfying constraints can be made tractable.

A quick, responsive system?. The correspondence between the organization of descriptions made in this representation and human perceptual organization neans that it is easy to "see" how to assemble a 3-D model. It also means that we try to modify or animate an existing model we will likely find that the changes we have to make are a simple function of the parameters of our model, rather than being, e.g., some hard-to-compute property of a collection of polygons or splines.

Because this part-based representation seems to have the right "control knobs" for manipulating 3-D models, it provides the basis for surprisingly effortless interaction: it took a moderately skilled operator less than a half-hour to assemble the

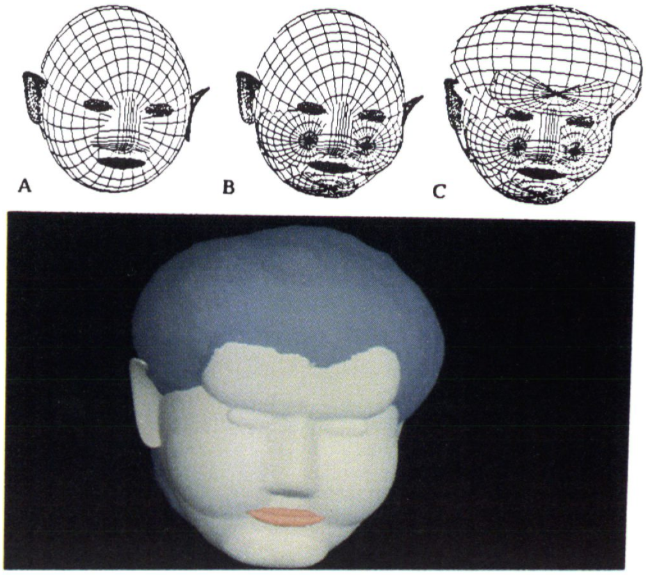

Figure 4. Building a face.

face in Figure 4, about five minutes to create the chair in Figure 1, and less than four hours each (including coffee breaks) to make the images in Figure 5. Much of this speed is due to the brevity of the final descriptions: to build the scene in Figure 5(a), for instance, requires positioning the mouse only 500 times

This performance is in rather stark contrast to more traditional 3-D modeling systems that might require several days to build up a complex scenes such as shown in Figure 5. This performance, perhaps more than any other statistic that could be given, illustrates how the close match between this representational system and the perceptual organization employed by human operators facilitates effective man-machine communication.

6 Summary

Man-machine interaction requires a representation that correctly describes the perceptual organization people impose on the stimulus. We have, therefore, presented a representation that has proven competent to accurately describe an extensive variety of natural forms (e.g., people, mountains, clouds, trees), as well as man-made forms, in a succinct and natural manner. The approach taken in this representational system is to describe scene structure in a manner that is like our naive perceptual notion of "a part," and to allow qualitative description of complex surfaces by means of physically- and psychologically-meaninful statistical abstractions.

To implement this system we have devised a user interface that allows the user to assemble forms in a natural manner, without having to be conscous of the details of either computer or program, and without having to move his hands unnessarily. This interface requires real-time feedback; to support this we have devised a linear-time hidden line algorithm that allows real-time display of two engineering views of the scene on a 68020-class machine without need for special hardware.

Each of the component parts of this representation — superquadric "lumps," deformations, Boolean combination, and the recursive fractal construction — have been previously suggested as elements of various shape descriptions, usually for other purposes. The contribution of this paper is to bring all of these separate descriptive elements together as a theory of human perceptual organization, and use them as the basis for man-machine interaction. In particular, we believe that the following are the important contributions this paper make toward solving the problems building and animating 3-D forms:

- We have demonstrated that this representational system is able to accurately describe a very wide range of natural and man-made forms in an extremely simple, and therefore useful, manner.
- We have found that descriptions couched in this representation are similar to people's (naive) verbal descriptions and appear to match people's (naive) perceptual notion of "a part."
- We have found that by using the fractal construction with various primitive elements and fractal scaling parameters we can mimic the sort of physically-meaninful statistical abstraction that people seem to employ when describing the shape of complex surfaces.
- And finally, we have shown that descriptions framed in the representation have markedly facilitated man-machine communication about both natural and man-made 3-D structures. It appears, therefore, that this representation gives us the right "control knobs" for discussing and manipulating 3-D forms.

Finally, however, we believe that the representational framework presented here is *not* complete. It seems clear that additional modeling primitives, such as branching structures [24] or particle systems [25], will be required to model the way people think about objects such as trees, hair, fire, or river rapids. Our future work will involve the integration of these primitives, together with time and motion primitives, into the framework that we have presented here.

REFERENCES

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

[2] Hobbs, J. (1985) Final Report on Commonsense Summer. SRI Artificial Intelligence Center Technical Note 370.

[3] Wertheimer, M. (1923) Laws of organization in perceptual forms, in *A Source Book of Gestalt Psychology,* W.D. Ellis (Ed.), New York: Harcourt Brace.

[4] Johansson, G., (1950) *Configurations in Event Perception,* Stockholm: Almqvist and Wiksell.

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

[6] Biederman, I., (1985) Human image understanding: recent research and a theory. *Computer Vision, Graphics, and Image Processing,* 32, 29-73

[7] Hoffman, D., and Richards, W., (1985) Parts of recognition, *Cognition* 18, 65-96

[8] 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.

[9] Prichard, R. M., (1961) Stabilized Images on the Retina, *Scientific American,* 9, 204, 72-79

[10] Piantinada, T. (1986) Illusory completion under conditions of visual stabilization, to appear, Science

[11] Konderink, Jan J., and van Doorn, Andrea J., (1982) The shape of smooth objects and the way contours end, *Perception, 11,* pp. 129-137

[12] Marr, D. (1982) *Vision,* San Fransico: W.H. Freeman and Co.

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

[14] Pentland, A. P., (1986) Perceptual organization and the representation of natural form, *AI Journal* 12, 1-34

[15] Gregory, R. L., (1970) *The Intelligent Eye,* New York: McGraw-Hill Book Company.

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

[17] Pentland, A. (1984d) Perception of three-dimensional textures, *Investigative Opthomology and Visual Science, 25,* No. 3, pp. 201.

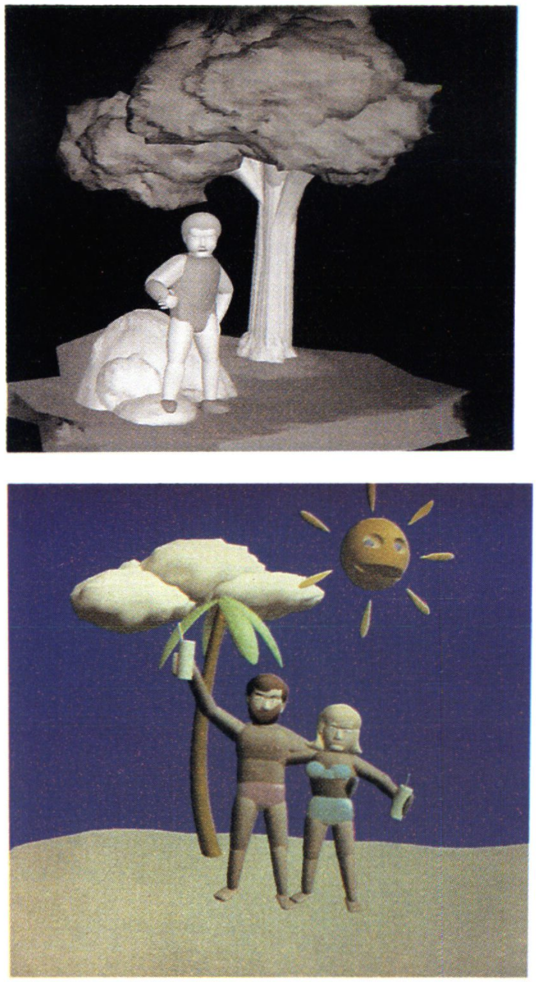

Figure 5. Two scenes that were both constructed in less than four hours, including coffee breaks, on a Symbolics 3600 Lisp Machine (equivalent to a 68020 class machine) without special equipment of any kind. (a) This scene has only 56 primitives (approximately 500 parameters/bytes of information), (b) only 100 primitives are required to model this scene (approximately 1000 parameters/bytes of information), despite the considerable detail in the faces. Both scenes contain about 40,000 polygons; color rendering time is 2.0 minutes.

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

[19] Gardener, M., (1965) The superellipse: a curve that lies between ellipse and rectangle, *Scientific American*, 9, 222-232

[20] Barr, A., (1984) Global and local deformations of solid primitives. *Computer Graphics 18,* 3, 21-30

[21] Ogden, J. M., (1985) Generation of fractals using the Burt pyramid, *Optical Society of America Annual Meeting* 14-18 October, p. 39

[22] Atherton, P. R., (1983) A scan-line hidden surface removal procedure for constructive solid geometry, *Siggraph '83,* 73-82

[23] Franklin, W. R., A linear time exact hidden surface algorithm *Siggraph '80,* 117-123

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

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