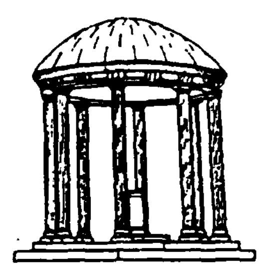
Grasping Reality Through Illusion
Interactive Graphics Serving Science

TR88-007

March 1988

N00014-86-K-0680

Frederick P. Brooks, Jr.



The University of North Carolina at Chapel Hill
Department of Computer Science
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175

Grasping Reality Through Illusion—
Interactive Graphics Serving Science

Frederick P. Brooks, Jr.

Department of Computer Science
University of North Carolina
Chapel Hill, N.C. 27599-3175

ABSTRACT

I treat three related subjects: virtual-worlds research—the construction of real-time 3-D illusions by computer graphics; some observations about interfaces to virtual worlds; and the coming application of virtual-worlds techniques to the enhancement of scientific computing.

We need to design generalized interfaces for visualizing, exploring, and steering scientific computations. Our interfaces must be direct-manipulation, not command-string; interactive, not batch; 3-D, not 2-D; multisensory, not just visual.

We need generalized research results for 3-D interactive interfaces. More is known than gets reported, because of a reluctance to share "unproven" results. I propose a *shells-of-certainty* model for such knowledge.

KEYWORDS: Interactive techniques, three-dimensional graphics, realism, human factors, simulation and modeling.

ILLUSION—SEEING UNSEEN WORLDS

The screen is a window through which one sees a virtual world. The challenge is to make that world look real, act real, sound real, feel real.

[Sutherland 65]

We graphicists choreograph colored dots on a glass bottle so as to fool eye and mind into seeing desktops, spacecraft, molecules, and worlds that are not and never can be. Ivan Sutherland's 1965 vision has driven the discipline's research program in the decades since. Viewing a current issue of the *SIGGRAPH Video Review*, visiting Video Night at SIGGRAPH, or riding StarTours at Disneyland shows how stunningly far we have come.

We must nevertheless ask, "Are graphics just for fun? Is all this technology for entertainment only? Or worse, for enthralling the mind while a sales message insinuates itself? Surely not!"

Even today computer systems help designers visualize electronic chips, mechanical objects, and buildings under design. This potent tool has equal promise in man's ongoing scientific enterprise—the understanding of the physical universe.

3-D Graphics. Some few virtual worlds can be idealized to be 2-D only, such as the Xerox PARC "desktop." Most worlds, including most objects in the physical world, are 3-D. Hence the virtual worlds research I shall discuss is all 3-D or more, and the user interfaces are 3-D.

Interactive Graphics. Attempting to make the view in the window seem real has produced a spectrum of research emphases. At one end, represented by the pioneering work of Turner Whitted, the effort has been to make the virtual scene *look* real, however long it takes. At the other end, represented at UNC by Fuchs, Pizer, and me, the effort has been to make the virtual scene *move* as if real, however sorry it looks. Then each effort works toward the other. I shall limit this discussion to real-time dynamic virtual-world systems.

Application to Scientific Modeling

The computational scientist, a metereologist, for example, builds mathematical models that describe the successive positions and interaction of real-world objects. The models calculate forces, energies, velocities, temperatures, charges, as a function of spatial position, and then calculate new positions for all the objects.

Just as it is useful to see a mechanical part or a building that has not yet been built, so it is illuminating to see

- how the scientifically modeled atmosphere will move and change as a result of the modeled processes, and
- some visualization of the spatial distribution of the (really invisible) temperature, pressure, wind velocity, electrical charge.

One wants to explore the virtual world resulting from such a calculation *interactively*—to choose where, for example, to study the weather in detail, or whether to look next at velocity or at pressure. One may want to release smoke at a point, for example, so as to see its stream line.

Moreover, one may want to choose interactively to which spatial regions one allocates scarce supercomputer time. Intermediate results may guide the interactive steering of the computation.

GRASPING—THE USER'S INTERFACE TO THE VIRTUAL WORLD

If indeed scientific computing can be made more insightful and efficient by interactive watching and steering, what should be the interfaces between the scientist and the virtual world he explores? How shall the scientist grasp the virtual objects? And how, by grasping objects, grasp concepts as well?

Our discipline is at the very beginning in answering these questions. What universals can we know about 3-D dynamic interfaces? Shall we just build systems, and report them in the Practice and Experience section of *Transactions on Graphics*? Or is interface design itself an area of research, producing generalizable results?

A Certainty-Shell Structure for Interface Knowledge—Any Data Are Better Than None

A major issue perplexes and bedevils the computer-human interface community—the tension between

- narrow truths proved convincingly by statistically sound experiments, and
- broad "truths," generally applicable, but supported only by possibly unrepresentative observations.

Some of us are scientists, insisting that standards of rigor be applied before new knowledge enters the accepted corpus. This insistence is the more vehement from psychologists, whose scientific discipline is young and whose rigor is hard-won. It is entirely proper.

Others of us are systems engineers, forced to make daily decisions about user interfaces and seeking guidance from any previous experience whatever. We observe that the issues in designing even one interface are many and large, and human factors studies can produce definitive answers to only a few questions at a time, and those narrow.

This tension colors all our communications. What papers shall we accept for this conference—results indisputably true but disputably applicable, or results indisputably applicable but perhaps over-generalized? Shall the anecdotal interface lore learned over decades of systems-building be relegated to unrefereed invited papers and keynote addresses?

In watching many awful interfaces being designed (and in designing a few), I observe that the uninformed and untested intuition of the designer is almost always wrong. We must always refine our interfaces by tests with real users, and we will always be surprised by those tests.

Previous experience on other interfaces does indeed help, however. Design principles can be induced and taught that will reduce design mistakes. *Over-generalized findings from other designers' experiences are more apt to be right than the designer's uninformed intuition.*

Or, put simplistically,

Any data are better than none.

A Proposal

How can this tension be relieved? I suggest that we as SIGCHI and as HFS define three nested classes of results—*findings*, *observations*, and *rules-of-thumb*.

Findings will be those results properly established by soundly-designed experiments, and stated in terms of the domain for which generalization is valid.

Observations will be reports of facts of real user-behavior, even those observed in under-controlled, limited-sample experiences.

Rules-of-thumb will be generalizations, even those unsupported by testing over the whole domain of generalization, believed by the investigators willing to attach their names to them.

Each of our conferences or journals should accept some reports of each kind. Referees and program committees must insist that results be correctly classified. Papers must be weighed for quality within the class to which they predominately belong. The appropriate criteria for quality will differ: truthfulness and rigor for *findings;* interestingness for *observations;* usefulness for *rules-of-thumb;* and freshness for all three.

To illustrate this proposal, I offer, as observations only, some lessons on 3-D interfaces from two decades of virtual-worlds research at Chapel Hill.

Virtual World Systems We Have Built*

Imaginative and ingenious virtual-worlds research has been done at several laboratories. Space does not permit a complete review. To describe the system space from which our observations come, however, here are sketches of fourteen experimental interactive 3-D virtual-world systems we have built at UNC. (Workers elsewhere have made systems similar to some of these.) They cover a wide spectrum: display technology, interface techniques, display tricks to get partial manipulability when complete manipulability is too hard, and total application systems.

*The videotape illustrating this section at the Conference will be submitted to the SIGGRAPH Video Review.

First, four examples of new display technologies:

PIXEL-PLANES. [Fuchs 85] A viewer is able to specify real-time dynamic motion to complex scenes of up to a total rate of 37,000 shaded, colored properly-hidden triangles per second. This 250,000-processor parallel engine, developed by Henry Fuchs and John Poulton, has enabled many of our later application systems.

Dynamic Varifocal Mirror System. [Hobgood 70, Fuchs 82] A radiation oncologist sees, on an oscillating-curvature mirror, a true 3-D point-cloud representation of the organ, the radioactive seeds he has planted in it, and the radiation isodose surface formed by the collection of seeds. Rotation (3-D), viewbox translation, and three pairs of viewbox clipping controls are all dynamic. Moreover, the oncologist, with no head gear, has all depth cues and can walk around or move his head for a different view.

Head-Mounted Display. [Holloway 87] The viewer, moving about in a 10-foot sphere, sees virtual objects (such as molecules) hanging in that space, optically superimposed on the real world and its objects. The superposition provides real scale and reference points to help in the study of the virtual molecule. The viewer can look from any direction and approach any part for more detailed study. Tracking head position and orientation swiftly and accurately is the challenging technical problem, not image generation.

GROPE. [Kilpatrick 76, Ming 88] A biochemist holding a remote manipulator docks a drug molecule, represented as colored, shaded spheres, into a protein similarly represented. As he moves the drug into the cavity, he feels, via motors on the manipulator, bump forces and electrostatic forces. Bumps are audible; their location is visually highlighted.

Besides the above experimental interfaces, two more requiring only standard technology:

FLASHLIGHT. [Holmes 85] A molecule, of hundreds or thousands of atoms represented as colored-shaded spheres, is viewed from a pre-specified viewpoint. The chemist can study its 3-D surface by pointing a real flashlight at the screen from any forward angle. The position of the flashlight is sensed by a videocamera. The light moves on each little sphere as the flashlight is moved, by color-table animation. The effect is as if the flashlight lit the model (except the light rays are parallel).

Head-Motion Parallax. [Fuchs 77] The chemist, wearing a small light on the forehead, views a screen scene. As the head moves from side to side, the scene counter-rotates to give proper head-motion parallax. A 1728-cell charge-coupled capacitor, shaded by a razor blade, is mounted above the screen. It senses the horizontal angular motion of the head. This effect is combined with standard rotating-shutter stereo.

Two techniques for generating complex images that are real-time manipulable:

FAST SPHERES. [Pique 82-83] On a standard Adage Ikonas display, the chemist can dynamically rotate and zoom molecules of a couple hundred atoms, represented as colored, shaded spheres. This was accomplished by ingenious microprogramming and judicious approximations. Today, thousand-atom spherical models are routinely manipulated on Pixel-planes.

Raster Molecule Docking System. [Palmer 87] The biochemist, working at an Ikonas terminal, docks a movable stick-figure drug molecule into a static protein represented by double-sized spheres. The surface defined by the spheres forms a very thin cavity. The biochemist fits the very thin stick figure fits into the cavity with the same constraints as if both drug and protein were represented by proper-sized spheres. Bump checking is easily done. (The technique was invented by David Barry of Washington University.)

Finally, six application systems:

GRIP Molecular Fitting System. [Britton 74, Britton 81] On a stereo vector display, the biochemist adjusts the atoms of a molecule into an optimum constrained fit against experimentally-determined electron density distributions. The density is represented by contours on the axial plane sets—"basket contours." Viewpoint, view distance, stereo disparity, and other viewing parameters are controlled independently of object manipulation. GRIP had the richest user interface we have yet built. Our users say it was the first molecular graphics system on which a new protein was solved entirely with virtual models, not brass ones. (Depending on how one defines priority, this was first accomplished either for Copper-Zinc Superoxide Dismutase [Richardson 77] or else for Erabutoxin. [Tsernouglou 77])

GRINCH Electron Density Interpretation System. [Williams 82] On a stereo vector display, the biochemist undertakes to find the protein's main-chain backbone in a new electron-density map, a task requiring rapid alternation between global and local views, and between coarse and detailed density representations. Density is represented by ridge-lines, which take on distinctive colors as the user assigns interpretations to them. The biochemist picks ridge-lines one after another, speaking their interpretations into a microphone.

Reciprocal-Space Diffraction Planner. [Harris 88] The crystallographer, planning how best to rotate the crystal in his diffractometer, moves a virtual surface in reciprocal space so as best to sweep out a virtual sphere in R-space, trading diffractometer time for selected redundancy. This ensures that the diffraction data—later Fourier-transformed into electron density—will be collected so as to optimize the density map quality.

Constructive Solid Geometry. [Godfeather 88] The designer of mechanical objects, working on two screens, sees the menu of primitives and the CSG tree of object relations on one screen. The other screen shows a lighted, shaded model of the ensemble, generated by Pixel-Planes. Blocks, cylinders, spheres, cones, toruses, and helicoids make up the primitives. Instances can be created, sized, colored, positioned and assembled instantaneously. Union, intersection, and subtraction are the relations. The image can be rotated and zoomed in real-time for viewing.

Anatomy Reconstruction. [Pizer 86] A radiologist studies for anomaly successive representations of a set of computer-assisted tomography scans. First, a stack of gray-scale images; then a stack of planes, each with density contours; then a wire-mesh figure, triangulated; then a set of surfaces for bones, organs, skin, delineated by lighting and transparency.

Perhaps for any one of the images, there is as an auxiliary view a 3-D entity, in two Euclidean dimensions and an abstract third dimension, showing progressive blurring, with the progressive extinction of small features. [Toet 86]

The radiologist can dynamically rotate and view any of these representations, and can cycle back and forth among them.

WALKTHROUGH [Brooks 86] The architect and/or client can walk through a virtual version of a building specified by plans and elevations. One screen shows position on the floorplan; the other shows the colored 3-D scene generated in real time by Pixel-Planes. Navigation may be by a pair of velocity-modulating joysticks (the helicopter metaphor), moving a 6-D sensor (the eyeball metaphor), or by walking on a treadmill steered with handlebars (the shopping-cart metaphor). The position of the sun and direct/ambient light ratio can be dynamically controlled.

Observations About 3-D Interfaces

3-D Understanding Is Difficult. People have rather precise world-models of their bedrooms or offices. We can navigate in the dark and reach for objects without looking at them. Forming similarly accurate mental models of virtual worlds requires hours of exposure to these worlds, plus every feasible cue.

Depth Cues. We observe depth cues to be effective in this order: hiding, kinetic depth effect, force cues, stereopsis, and others. [Kilkpatrick 76] Perspective is very effective when parallel lines and right angles abound in the model, but it can even be counterproductive when the discovery of parallel elements amidst disorder is part of the user task, as it is with molecular structures.

Stereopsis, on the other hand, is not especially potent when strong perspective cues are already present. When, as with molecules, perspective cues are not very helpful, our users find steropsis to be very powerful. Stereopsis works best when the viewer can smoothly increase disparity from zero while looking at the scene.

Exploratory Viewing. The kinetic depth effect is very powerful. We were surprised to see it effective even when a complex molecular image is rotated as slowly as one update (jump) per second. The ability to move viewpoint therefore needs to be a separate control in most interfaces. [Lipscomb 81] (Rocking is much better than rotation.)

Viewpoint specification requires 6 degrees-of-freedom in general, although some of camera roll, pitch, and yaw can sometimes be defaulted. In the seashell metaphor, for example, one studies an upright object centered at the coordinate origin as one does a seashell in the hand: two view-from angles and a viewpoint distance suffice.

User-positioned light sources and user-controlled camera zoom substantially enhance the perception of structure in the exploration of a passive virtual scene.

Eight of our fourteen virtual-world systems provide only exploration of passive scenes—no manipulation. This mode would seem to satisfy many scientific visualization needs. It is time to systematize the knowledge required for so simple a class of interfaces.

Map versus Scene Navigation. The user often needs to see both a view of the virtual world and a map showing where he is and where he is looking. Again and again we have evolved to using two screens, or a scene screen that is inset with an auxiliary map view. I think it sensible to plan such from the start. One needs a separate map for each continuous parameter space that controls the scene. It is handy for the user to be able to attach labels to points in these maps.

We observe that viewers start out doing map navigation and then progress to scene navigation as they build a precise mental world-model. This corresponds, of course, to real-world behavior.

The two kinds of navigation require different metaphors, and different interface devices are appropriate. In WALKTHROUGH, for example, the Eyeball 6-D cursor was ideal for map navigation but quite confusing for scene navigation via a projector screen. Turning the Eyeball to change direction misaligns its coordinates relative to the scene coordinates.

Progressive Refinement. Real-time motion, complex world-models, and high-quality images create a workload that overwhelms today's hardware. I think that will always be true. We find that a dynamic technique, rather than a static compromise, best resolves the dilemma:

- always move objects realistically, no matter what else suffers.
- sacrifice image resolution or model complexity or image quality while the user is moving objects.
- as soon as the user stops moving objects, automatically invoke progressive image-refinement, resolution improvement, progressive detailing, or anti-aliasing.

By this method, the user gets a high-quality picture within seconds after motion stops. The Varifocal Mirror system cuts resolution ten-fold during motion. Pixel-Planes does anti-aliasing as soon as viewpoint motion ceases.

The picture must never jump or move discontinuously. Even an occasional jump destroys the illusion of reality one has labored so hard to create.

Realistic illusion of motion requires not only rapid update rates (20 to 30 updates/second), it also requires very short lag between the action of a dynamic device and its effect on the view. Lags show up sharply when one is trying to hold a virtual object still in space as one moves the head about it.

Bump Detection. In the real world, two objects cannot occupy the same space at one time, nor pass through one another. Virtual-world researchers have almost accustomed themselves to accepting that computer graphics worlds have the opposite property, because most do. We must not surrender so easily. Space-exclusion plays a surprisingly large role in our perception of "realness." Difficult though it may be, we must keep bump-checking high on our agendas.

Multisensory Interfaces. Although vision is the dominant sense in humans, we find visual illusions to be much enhanced by reinforcing illusions to other senses.

The GROPE arm puts out forces that keep one object from intersecting another. The hard-surface illusion is enhanced by an audible click when a virtual bump occurs.

Moreover, even if no force display is provided, we observe propriopositional effects to reinforce visual illusions. For example, if two chemists are looking at the same molecule, the one whose hand is on the viewpointer seems to get a stronger kinetic depth effect as the viewpoint is moved.

We also find the kinesthetic experience to enhance the visual illusion in the cases of the 6-D Eyeball that one positions in building floorplan space, the flashlight one physically moves in front of a molecule picture, and the treadmill with which one walks in scene space. The ability to move objects, not just viewpoints, in the virtual world appears to go yet further in enhancing perception of that world. [Brooks 77]

Obviously, cost will retard the widespread fielding of such interfaces. Nevertheless, we researchers need to understand what works, what does not help much, and why. Then we need to distinguish in design between the ideal assignment of devices to input parameters and the compromises forced by cost.

Metaphor Matters. In designing interfaces we find that the explicit selection of a metaphor for each interface substantially helps us in defining the issues and making consistent decisions. Different metaphors demand, or are permitted by, different classes of display devices and input devices. In WALKTHROUGH, for example, screen projection constrains us to metaphors with look-ahead and no real-world motion, such as a helicopter metaphor, or a steerable shopping cart. A head-mounted display would permit look-aside and a tethered real-world motion, rather like walking about on a separately navigated flat-bed truck. Metaphor determines whether position or velocity-specifying devices are appropriate.

Direct Manipulation versus Menus versus Command Strings. We distinguish the discrete *interactive* change of a virtual world parameter from the continuous *dynamic* change of parameters. Dynamic actions should be specified by dynamic devices; interactive commands by menu selection. [Britton 81] The only proper use of a typed character command in a virtual-world system is to call up by name some object from the database, or to attach a name to an object.

Many Dynamic Input Variables. I have been surprised to find that virtual-world systems for real applications always require many more dynamic input variables than we expected. One instinctively expects about 6 dynamic degrees-of-freedom to suffice. The GRIP molecular fitting system, for example, has 22 such, mapped onto 15 joysticks, sliders, and dials. Six of these variables position and orient an amino acid in its density, 8 more set its side-chain torsion angles. View direction, stereo disparity, rocking rate, and other viewing parameters account for the rest. The Walkthrough system has 10 dynamic input parameters; the GROPE molecule docking system, 15; the Varifocal mirror system, 13.

Kinesthetic Selection versus Tactile Continuity. We find users are best able to control lots of dynamic variables when each set of correlated variables (such as x, y, z translation) is mapped onto a single, separate input device. Users then find most of the devices by feel, without losing visual continuity. [Lipscomb 81] This mapping has been much more effective than preserving tactile continuity by overloading definitions for one device, such as a mouse. When cost forces this latter compromise, it becomes desirable to divorce the hand-cursor (mouse) from the screen cursors, and leave sticky screen cursors visible in each parameter space. Screen cursors must always move in directions consistent with hand motion. (Users easily get used to hand-away=screen-up, but any other direction shifts are confusing.)

The Two-Cursor Problem. Videotapes of menu users show a recurring pattern: operand-pick, command-pick; operand-pick, command-pick. The specification of commands interrupts both the visual and tactile continuity inherent in the operand cursor's natural movement. One needs two cursors—an operand cursor that moves continuously in the viewbox, and a command cursor that jumps discretely among the commands.

Many solutions suggest themselves: two cursors, with left- and right-hand mice; pop-up menus (a palliative, not a cure); a foot mouse, etc. One of the best solutions is the Macintosh's—command keys for the left hand while the right one mouse-selects operands. The elegance of this is that the command keys mirror menu items, so one gets menu-provided prompting for less-familiar commands, prompt-free continuity for more familiar commands, and smooth incremental enlargement of the fluent vocabulary.

Command selection is a natural candidate for segmented speech recognition. Vocabularies are limited, utterances are naturally segmented, menus prompt for the standardized vocabulary. The technology for speaker-specific segmented speech recognition is available off-the-shelf.

Our experiments with a Votan system as command recognizer for the GRINCH electron-density interpreting system show this approach to have much promise. Recognition rates are better than 95%; pick time is about the same as with tablet, and less than one second; training the system takes about 20-30 minutes for a new user. We do not have any data on over-all effectiveness.

REALITY—SCIENTIFIC COMPUTING AND THE MODELING OF REAL WORLDS

The purpose of computing is insight, not numbers.
[Hamming 62]

The glory of the physical sciences in the period from Newton through Einstein was the development of mathematical models and the mathematics (principally calculus and statistics) with which to analyze them. In the same way, the past forty years have seen the development of models, of mathematical tools, and of mathematical machines that have vastly expanded our modeling power.

The computing power available to the working scientist has exploded beyond our ready comprehension. The Mac on my desk is faster (fixed-point only) and has more memory than the IBM Stretch supercomputers, the world's largest and fastest from 1961-65. My access is continuous and interactive, not via batch queue. The effect of such desktop machines on the immediacy of scientific computing is incalculable.

At the other end of the spectrum, today's supercomputers are about 1,000 times faster than Stretch, with 1,000 times as much memory. Networks of workstations offer even more cycles, even more memory.

Today such computers empower us to build

sophisticated models
of
complex natural phenomena

and to **explore** them for new **insights** into models and phenomena.

Sophisticated mathematical models
massive detailed 3-d non-linear discontinuous discrete indeterminate parametric parametric symbolic

Figure 1

These powerful models undertake descriptions inconceivable for closed-form algebraic solutions from continuous mathematics. The consequences of theory can be explored as never before, and as those consequences are tested against data, the theoretical models are rapidly being made yet more faithful and yet more ambitious. Table 1 shows some of the attributes of today's models.

These models, in turn, allow the study of natural phenomena always known to be complex. Table 2 shows some of these.

For some models, hours of computation yield a handful of results, easily compared against experimental values. Computational quantum chemistry, for example, undertakes *ab initio* solutions to Schrödinger's wave equation. The models are tested and refined by calculating known constants, such as the mass of the electron.

For many other massive computations, however, the volume of the results increases with the size and resolution of the models. Modern weather modeling, for example, covers large areas with high resolution—producing millions of output values.

The power explosion in scientific computing has itself created a new crisis: How shall these results be understood? How can we produce insight, instead of just numbers?

Complex natural phenomena
species population dynamics protein structures magnetohydrodynamics plate tectonics of the earth shock-wave propagation quasars, pulsars, black holes particles and quarcks hemisphere metereology oil-field geology blood flow in the body

Figure 2

If mathematics is queen of the sciences, computer graphics is the royal interpreter. The data gleaned from the real world and models of it can best be translated to human insight by being cast as pictures. [McCormick 87]

Likewise, the queries and orders to our mathematical models are often best transmitted sub-verbally by picking, poking, or pushing virtual objects.

Graphics for Insight versus Graphics for Publication

If one looks closely at today's use of computer graphics as handmaiden to large-scale scientific computing, several facts emerge:

- Visualizations are today more often used for communicating the investigator's insights to others than for generating insights. Indeed, an oft-voiced request is for "pictures I can show my funding agency."
- Visualizations are made after-the-fact, when the computation is complete, rather than interactively while it proceeds.
- Visualizations are rarely used to communicate insights in mid-flight so that the investigator can *guide* the computation.
- Setting up program parameters to get effective graphics output is hard, time-consuming work. Motion sequences are *very* burdensome to specify.

Graphics Is Hard Work. The last of these observations explains the first three: getting insight-producing graphical output from a scientific computation is today just plain hard—too hard to be used routinely.

We graphicists and interface designers must produce generalized graphics packages so helpful, so adaptable, so easy to use that our scientists will use them as routinely as they today use computer text-processing to write their papers.

It is easy for us to fool ourselves into thinking we have achieved this goal, for the highly-motivated scientist (e.g., a doctoral student) will master any tool, no matter how awkward. So let me share what we have found to be essential:

The Chapel Hill Criterion (CHiC):

Our systems must be:

- *so simple full professors can use them, and*
- *so fruitful that they will.*

Looking Pretty vs. Looking Insightful. When graphics are produced primarily for publication, one naturally works hard on their attention-getting and esthetic attributes—making pretty pictures. That is an art computer graphicists have cultivated assiduously.

As we increasingly make it our aim to produce insight in the minds of the scientific investigators themselves, we shall have to turn our ingenuity to a different kind of challenge. We shall have to learn some perceptual psychology so as to know what communicates. More important, we shall have to exercise our imaginations as to how we should like to see our data if we had magical technology.

Many Visualizations versus One. Sutherland's challenge, to make the view in the window look real, although hard to meet, is easy to state. It applies only to virtual objects that have real visible counterparts. Sutherland's first system, for example, was for the design of mechanical parts. We can readily judge how nearly a steel machining in the display window looks like a steel machining in the hand.

What Does a Molecule Look Like? For molecules, galaxies, electric fields, and stress distributions, there is no real "look." So there can be no single criterion for the success of the visualization. Indeed, we have found different visualizations of the same molecule each to produce a different insight. [Pique 82]

That view is best which provokes the most profound insight; who is to say in advance which it will be? It may well be neither the most detailed nor the most highly resolved. The more visualizations one sees, the better the chance of finding a fruitful one.

The best visualization strategy for abstractions is therefore quite different than for real objects. Rather than working on making one visualization ever closer to the ideal, the view-maker should devote his energies to the production of *many different* visualizations, as many as imagination will conceive and energy will permit. Our goal, then, must be to give scientists tools to easily generate many insight-producing visualizations from the same data set.

VIEW—the Visualization Impromptu Evaluation Workbench

Our GRIP molecular graphics team in Chapel Hill is building a prototype of such a system. The VIEW workbench will initially be specialized for the generation of visualizations of protein and nucleic acid molecules, for that is the scientific computation we know best. And we are firmly convinced that any technology, including interactive computer graphics, advances fastest when coupled to real users and focused on a driving problem.

In the VIEW system, chemists will call up molecular data from standard databases using the off-the-shelf Mendyl system produced by Evans & Sutherland's Tripos Associates Division. Then they will choose among several geometric or abstract representations of the structure: stick-figure, sphere collection, ribbon, solvent-accessible surface, density cloud, reciprocal-space plot. Atom coordinates will be used to build this structure, the skeleton of the visualization.

Other variables from the database (e.g., atom type, charge distribution, temperature factors) can then be mapped, by impromptu selection, upon the other visual carriers: color, transparency, intensity, etc. We are not aware that the extemporaneous mapping of database "freight" onto graphics "carriers" has been attempted before.

The user will then have a structured 3-D visualization. It can be dynamically viewed from any viewpoint, have its color tables dynamically changed, and be otherwise interactively explored. When a particular visualization has been polished and explored, the user can save it, photograph it, videotape it from standard camera paths, or scrap it entirely and start over with another visualization concept. The overall technical challenge, then, is to push spontaneous interaction a whole level back, into visualization *formulation* itself.

We expect a skeletal prototype to be working in March, and a prototype useful to biochemists by June.

In-Flight Watching of Computations

An investigator would get more insight from many computations if one could see intermediate results just as quickly as one could absorb them. Yet I know of no substantial supercomputer application that is run this way.

Why is this so? Well, convenience is one reason. Watching a computation from 3:00 to 5:00 a.m. seems somehow less attractive than perusing selected results by daylight. Moreover, the problem owner has to do it in person; one cannot send a graduate student.

Difficulty is another reason. The richest insights require interactive selection of what will be viewed, depending upon how the computation flies. We do not yet have the computer graphics or data capture tools to make impromptu visualization easy.

One suspects, however, that the modesty of the effort devoted to the interactive observing and steering of supercomputer calculations is due as much to ideology as to difficulty. One hears two arguments from supercomputer proprietors:

- "We are saturated and can't afford the time required to dump all those intermediate results and/or to calculate the visualizations."
- "We are committed to giving as good service to network users as to local users. Such interactive monitoring and steering either requires networks faster than exist, or favors the local user."

As to the first, the purpose of computing is insight. The only valid question is whether the cycles used for visualizing the computation produce more insight than more raw grinding would produce. That judgement cannot be made without experiments and experience. I would argue that the case for visualization is *prima facie* plausible enough to justify massive experiments.

In some centers, funds that could have been used to expand the computer itself are being used instead to buy a mini-super, broadband-attached, which does the visualizations and all other user interactions. That makes a lot of sense to me.

As to the equal-remote-service argument, I think that it is dangerous hogwash. We must not achieve equality of remote service by foregoing the best possible local service. The whole nation can only lose by that policy.

Let us press forward with broadband networks, by all means. At the same time, we must enhance the local user's power as much as possible. That means interfaces for interactive modeling. After all, the remote user can *become* a local user by travel, as one does for astronomical observatories and particle accelerators. The bandwidth of an airplane carrying an investigator, his brain, and his magnetic tape is very high.

In-Flight Interactive Steering of Computation

Most computations, even on supercomputers, are so short that interactive watching and steering make no sense. From earliest times, however, some computations have been run for hours, days, or weeks. Although few in number, these jobs consume a significant fraction of all scientific computer cycles.

Some of these big jobs are in fact single, monolithic computations. Many others are searches in some parameter space—drug conformations are exhaustively tested, chip floor-plans are generated by simulated annealing, optimum operating points are found by hill-climbing.

This is true not only for massive computations: many short computations are episodes in a long series which itself is a search in a parameter space. Between each shot the investigator refines his program, his numerical method, or his underlying model. The intellectual activity of scientific computing is the same as that of physics itself:

- modeling of nature, whose complexity is in fact everywhere dense,
- testing the model against experimental data,
- modifying the model, until it explains nature "well enough."

So it is that much "debugging" is really model refinement, and many scientific programs run successfully only once. Corollary: many, perhaps most supercomputer cycles are used on wrong parameter regions or wrong models, and could be eliminated with no loss to science. The only problem is to identify which ones, *before* they are computed.

The investigator can usually do a *far* better job of pruning and directing the search than any pre-programmed test can, just as a geologist will collect a more interesting bushel of moon-rocks than will a programmed robot:

- The investigator knows the problem, the entire global context in which to evaluate intermediate results.
- The investigator can recognize patterns that have occurred before when the search was in unfruitful alleys.
- The investigator is free to bet on informed hunches, a freedom he would not entrust to a blind program.

If, then, we go beyond in-flight watching to in-flight steering, we can hope to save machine cycles wholesale—enough to pay for the insight-producing watching, and more besides.

I do not propose that the calculation should stop and wait for user guidance. In multi-hour calculations, loose-coupled guidance lagging 5-15 minutes behind the computation can nevertheless obviate the exploration of many blind alleys. "It is more important to do the right computation than to do the computation efficiently."

Tools for Interactive Watching and Steering

Computational scientists, computer graphicists, and interface designers need to address generalized tools for interactive steering of large computations. Lipscomb suggests that the traditional interactive symbolic debugger is the proper platform on which to build supercomputer matching and steering tools. Such tools must be universal and application-independent. Symbolic debuggers are. More important, such tools must demand almost no cooperation from the supercomputer program, which is often written, modified, or used by one, or a few researchers part-time. Traditional symbolic debuggers require only a recompilation with the appropriate option and no special, hand-crafted arrangements in the program source. Interactive watching and steering must aspire to that standard.

This basic software technique must be combined with modern graphics techniques to permit the quick and flexible selection of visualization tools. It must be possible then to bind a visualization script to the application's data files so that it can be invoked for a later computation run, or even for a separate visualization exploration run.

The range of visualizations allows invention without bound. A vendor-supplied toolkit is not enough. There must be hooks on which the computational scientist or his collaborating graphicist can hang visualization tools they write themselves.

Acknowledgements

These ideas were formed over a decade of collaboration with Michael Pique, now of the Scripps Research Institute, and James S. Lipscomb, now of the IBM Thomas J. Watson Research Center. They have been refined by discussions with my UNC colleagues Henry Fuchs, Stephen Pizer, Helga Thorvaldsdottir, Turner Whitted, David Beard, James Coggins, and our graduate students. David Lines carefully prepared the manuscript.

My teams' work has been supported by the NIH Division of Research Resources, NSF, ONR, and Burroughs-Wellcome. The work reported here and done by colleagues' teams has been supported by DARPA, NSF, the NIH National Cancer Institute, the NIH Stroke Institute, and ONR,

References

Bergman, L., Fuchs, H., Grant, E., Spach, S. [1986] Image rendering by adaptive refinement. *Computer Graphics: Proceesings of SIGGRAPH '86*, 20, 34 (August), 221-240.

Britton, E.G. [1977] *A methodology for ergonomic design of interactive computer graphics systems, as applied to crystallography.* Ph.D. dissertation, Universtity of North Carolina, Chapel Hill.

Britton, E., Lipscomb, J.S., Pique, M., Wright, W.V., and Brooks, Jr., F.P. [1981] The GRIP-75 man-machine interface. Videotape presented at SIGGRAPH Conference.

Brooks, Jr., F.P. [1977] The computer "scientist" as toolsmith: studies in interactive computer graphics. In *Information Processing 77* (Toronto). North-Holland, Amsterdam, pp. 625-634.

Brooks, Jr., F.P. [1986] Walkthrough—a dynamic graphics system for simulating virtual buildings. In *1986 Workshop on Interactive 3-D Graphics* (Chapel Hill).

Fuchs, H., Duran, J., and Johnson, B. [1977] A system for automatic acquisition of three-dimensional data. In *National Computer Conference*, 49-53.

Fuchs, H., Pizer, S.M., Heinz, E.R., Tsai, L.C., and Bloomberg, S.H. [1982] Adding a true 3D display to a raster graphics system. In *IEEE Computer Graphics and Applications* 2, 7 (September), 73-78.

Fuchs, H. and Abram, G.D. [1985] Specialized computer organization for raster-graphics display. In *Algo-rithmically-specialized Parallel Computers*, edited by L. Snyder, L.H. Jamieson, D.B. Gannon, and H.J. Siegel. New York: Academic Press.

Fuchs, H., Goldfeather, J., Hulquist, J.P., Spach, S., Austin, J.D., Brooks, Jr., F.P., Eyles, J.G., and Poulton, J. [1985] Fast spheres, shadows, textures, transparencies, and image enhancement in Pixel Planes. *Computer Graphics* 19, 3 (July), 111-120.

Goldfeather, J., Molnar, S., Turk, G., and Fuchs, H. [1988] Near real-time CSG rendering using tree Normalization and geometric pruning. UNC Technical Report.

Hamming, R.W. [1962] *Numerical Methods for Scientists and Engineers.* New York: McGraw-Hill.

Harris, M.R. [1988] An introduction to reciprocal space modeling and rspace. UNC Technical Report.

Hobgood, Jr., William S. [1970] A 3-D computer graphics display using a varifocal mirror. M.S. thesis, UNC.

Holloway, R.L. [1987] Head-mounted display technical report. UNC Technical Report.

Holmes, David H. [1985] Three-dimensional depth perception enhancement by dynamic lighting. M.S. thesis, UNC.

Kilpatrick, P.J. [1976] *The use of a kinesthetic supplement in an interactive graphics system.* Ph.D. dissertation, University of North Carolina, Chapel Hill.

Lipscomb, J.S. [1981] *Three-dimensional cues for a molecular computer graphics system.* Ph.D. dissertation, University of North Carolina, Chapel Hill.

Lipscomb, J.S. and Brooks, Jr., F.P. [1981] The GRIP-75 man-machine interface (videotape). In *ACM SIGGRAPH Video Review*, 4 (August).

McCormick, B.H., DeFanti, T.A., and Brown, M.D., eds. [1987] *Visualization in Scientific Computing.* Report of the NSF Advisory Panel on Graphics, Image Processing and Workstations.

Ming, O.Y., Pique, M., Hughes, J., Brooks, Jr., F.P. [1988] Using a manipulator for force display in molecular docking. IEEE Robotics and Automation Conference.

Palmer, T.C. [1987] Docktool: a dynamically interactive raster graphics visualization for the molecular docking problem. M.S. thesis, UNC.

Pique, M., Richardson, J.S., and Brooks, Jr., F.P. [1982] What does a protein look like? Videotape presented at SIGGRAPH Conference.

Pique, M. [1982-83] Fast 3-D display of space-filling molecular models (videotape). In *SIGGRAPH Video Review* 6 (October, 1982), 10 (October, 1983).

Pique, M. and Brooks, Jr., F.P. [1985] Computer graphics for molecular modeling. In *Molecular Dynamics in Protein Structure* Edited by J. Hermans.

Pizer, S.M., Fuchs, H., Mosher, C., Lifshitz, L., Abram, G.D.,Ramanathan, S., Whitney, B.T., Rosenman, J.G., Stabb, E.V., Chaney, E.L., and Sherouse, G. [1986] 3D shaded graphics in radiotherapy and diagnostic imaging. In *Computer Graphics Proceedings*, 107-113

Richardson, D.C. [1977] Three-dimensional Structure of Cu, Zn, Superoxide Dismutase. *Superoxide and Superoxide Dismutases.* Edited by A.M. Michelson, J.M. McCord, and I. Fridovich. New York: Academic Press, 1977.

Sutherland, I.E. [1965] The ultimate display. *Proceedings of IFIP 65*, vol. 2, pp. 506-508, 582-583.

Toet, A., Koenderink, J.J., Pizer, S.M., Helmink, L. [1986] A multiresolution image segmentation algorith based on a topological description of image structure. *First Quinquennial Review*, Utrecht.

Tsernoglou, D. and Petsko, G.A. [1977] Three-dimensional-structure of neurotoxin a from venom of the Philippines sea snake. In *Proc. National Academy of Sciences USA* 74, 3 (March), 971-974.

Tsernoglou, D., Petsko, G.A., McQueen, Jr., J.E., and Hermans, J. [1977] Molecular graphics: application to the structure determination of a snake venom neurotoxin. *Science* 197, 4311 (30 September), 1378-1381.

Williams, T.V. [1982] *A man-machine interface for interpreting electron-density maps.* Ph.D. dissertation, University of North Carolina, Chapel Hill.