The Computer "Scientist" as Toolsmith-
Studies in Interactive Computer Graphics

TR88-041

August 1988

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

First appeared in Information Processing 77, B. Gilchrist, ed., pp. 625-634
North Holland Publishing Company, 1977

THE COMPUTER "SCIENTIST" AS TOOLSMITH—STUDIES IN INTERACTIVE COMPUTER GRAPHICS*

FREDERICK P. BROOKS, JR.
University of North Carolina
Chapel Hill, North Carolina

(INVITED PAPER)

Computer "scientists" are in fact "engineers of abstract objects," as Zemanek says. Moreover, we are toolsmiths, building for others to use.

This viewpoint has driven a decade of work in interactive computer graphics at our laboratory. I review six of our major steps and summarize their scientific results and engineering lessons.

Our latest tool, for manipulating protein and nucleic acid molecules, is now producing publishable chemical results for several client teams. We find clients to vary substantially in their preferences among various perceptual and manipulative aids.

Our experience strongly supports Alexander's thesis that good fit cannot be directly defined or designed; it is the absence of misfit, achieved by iterative design.

1. THE COMPUTER "SCIENTIST" AS TOOLSMITH

1.1 The name and nature of computer science

When our discipline was new-born, there was the usual perplexity as to its proper name. We at the University of North Carolina, like many others, settled on computer science as our department's name. Now, with the benefit of a decade's hindsight, I believe that to have been a mistake, and I believe it important to understand why, for we will better understand our task.

What is a science? Webster says,

> "A branch of study concerned with the observation and classification of facts, especially with the establishment and quantitative formulation of verifiable general laws."

This puts it pretty well -- a science is concerned with the discovery of facts and laws.

A folk-adage of the academic profession says, "Anything which has to call itself a science isn't." By this criterion, physics, chemistry, geology, and astronomy may be sciences; political science, military science, social science, and computer science are not.

Perhaps the most pertinent distinction is that between scientific and engineering disciplines. That distinction lies not so much in the activities of the practitioners as in their purposes. A high-energy physicist may easily spend most of his time building his apparatus; a spacecraft engineer may easily spend most of his time studying the behavior of materials in vacuum. Nevertheless, the scientist builds in order to study; the engineer studies in order to build.

What is our discipline? I submit that by any reasonable criterion, the discipline we call computer science is in fact not a science but an engineering discipline. We are concerned with making things, be they computers, algorithms, or software systems.

Unlike other engineering disciplines, much of our product is intangible: algorithms, programs, software systems. Prof. Heinz Zemanek has aptly defined computer science as "the engineering of abstract objects." [1] Even when we build a computer, the computer scientist designs only the abstract properties, its architecture and implementation. Electrical and refrigeration engineers design the realization.

In contrast with many engineers who make houses, cars, medicines, clothing for human need and enjoyment, we make things that do not themselves meet human needs, but serve as tools in the meeting of needs. In a word, the computer scientist is a toolsmith -- no more, but no less. It is an honorable calling.

If we perceive our role aright, we then see more clearly the proper criterion for success: a toolmaker succeeds as, and only as, the users of his tool succeed with his aid. However shining the blade, however jeweled the hilt, however perfect the heft, a sword is tested only by cutting. That swordsmith is successful whose clients die of old age.

1.2 How can a name mislead us?

If our discipline has been misnamed, so what? Surely computer science is a harmless conceit. What's in a name? Much. Our self-misnaming hastens various unhappy trends.

First, it implies that we accept a perceived pecking

*Our graphics research has been supported by AEC contract AT-(40-1)-3817, NSF grant PJ34697, N.I.H. grant RR 00898, U.S. Army Research Office contract DAAG29-70-G-0240 and the IBM Corporation.

First appeared in: Information Processing 77, B. Gilchrist, ed., pp. 625-634 Copyright IFIP, North-Holland Publishing Company (1977)

order that respects natural scientists highly and engineers less so, and that we seek to appropriate the higher station for ourselves. That is a self-serving gambit, hence dubious. It is also a risky gambit; in the case of some upstart social "sciences" the name is merely ludicrous and makes the practitioners look foolish. Moreover, the gambit is futile -- we shall be respected for our accomplishments, not our titles.

Second, sciences legitimately take the discovery of facts and laws as a proper end in itself. A new fact, a new law is an accomplishment, worthy of publication. If we confuse ourselves with scientists, we come to take the invention (and publication) of endless varieties of computers, algorithms, and languages as a proper end. But in design, in contrast with science, novelty in itself has no merit. If we recognize our artifacts as tools, we test them by their usefulness and their costs, as is proper.

Third, we tend to forget our users and their real problems, climbing into our ivory towers to dissect tractable abstractions of those problems, abstractions that may have left behind the essence of the real problem.

We talk to each other and write for each other in ever more esoteric vocabularies, until our journals become inaccessible even to our society members, and publication properly commands a higher price from the author in page charges than from the reader in subscription fees. So our writings even in their economics resemble garbage, for which the generator pays the collector, rather than vice versa.

This deadly trend already curses American mathematics; its cold chill can be felt in computer science. We are succumbing to the occupational illness of scholars diagnosed 2000 years ago by Our Lord Jesus Christ: "You desire praise from one another." [2]

Fourth, as we honor the more mathematical, abstract, and "scientific" parts of our subject more, and the practical parts less, we misdirect the young and brilliant minds away from a body of challenging and important problems that are our peculiar domain, depriving these problems of the powerful attacks they deserve.

These are the system design problems characterized by arbitrary complexity. Examples are the intricate demands of business data processing or of operating systems.

These problems scandalize and discourage those who approach them from backgrounds of mathematics and natural science, and for different reasons. The mathematician is scandalized by the complexity -- he likes problems which can be simply formulated and readily abstracted, however difficult the solution. The four-color problem is a perfect example.

The physicist or the biologist, on the other hand, is scandalized by the arbitrariness. Complexity is no stranger to him. The deeper the physicist digs, the more subtle and complex the structure of "elementary" particles he finds. But he keeps digging, in full faith that the natural world is not arbitrary, that there is a unified and consistent underlying law if he can but find it.

No such assurance comforts the computer scientist. Arbitrary complexity is our lot, and here more than anywhere else we need the best minds of our discipline fashioning more powerful attacks on such problems.

What name should we bear? I am not hopeful that our established name will be changed, and I think the need hardly worth a crusade. Hence the purpose of these observations is not to suggest a renaming movement, but to raise conscious mental defenses against the subconscious attitudes. The most important of these defenses are a continual focus on our users and a continual evaluation of our progress by their successes.

2. INTERACTIVE COMPUTER GRAPHICS RESEARCH

Since 1967, our laboratory has been focussing on toolsmithing and users by studies evaluating the effectiveness of interactive computer graphics (ICG) in real applications and developing better graphics techniques. Many investigators have been working on ICG in many laboratories; space does not permit me to treat other work in the field. Much of our own work has been previously reported; in this paper I give merely an overview of our decade's work and summarize the scientific results and the engineering lessons learned.

2.1 Why study interactive computer graphics?

Why should research computer scientists want to serve as graphics toolsmiths? I see four distinct reasons:

1. The whole discipline of system architecture has as its central concern the definition of the interface between a computer system and its users. In interactive computer systems, the smoothness of the interface is even more crucial than in batch systems. An interactive system, therefore, constitutes a test-bed for studying demanding interface requirements.

2. An ICG system provides the raw materials from which a broadband man-machine interface can be built. In an ICG system we can hope that equipment awkwardnesses will be sufficiently removed that we can observe and study the more fundamental psychological aspects of the interface. In essence, ICG systems constitute the cutting edge of all close-coupled man-machine systems.

3. ICG systems are simple prototypes of distributed-intelligence systems, a family of designs we need to understand better. Because CRT images demand continual refreshing if they are to represent continuous motion, ICG systems typically involve at least two processors, one to refresh the image and perhaps transform it, and another to perform the desired manipulations on the model underlying the display. Many systems, including ours, also time-share the services of a more powerful computer for hard-work calculations. In this simple but real-time context, the problems of division of labor, synchronization, and intermachine communication that arise in all distributed intelligence systems can be studied.

4. Accomplishment in toolmaking can take two forms.

One can design a tool that achieves commercial success and is manufactured in quantity. Satisfaction then comes from the aggregate usefulness of all the copies. Alternatively, a tool may have sufficient power that each copy achieves a high degree of new usefulness. ICG offers the latter kind of challenge and satisfaction. We select applications where brute-force computing is currently hopeless, where human pattern recognition, global context perception, and experienced judgement are needed to guide a computation. But these applications also have the property that unaided human skills are not enough, where computation is inherent to the process. In such applications, the computer acts as an intelligence amplifier for the user. Since first-class, trained, experienced intelligences are scarce in any field of endeavor, amplifying their effectiveness and productivity can yield high reward.

2.2 What does interactive computer graphics research take?

Most ICG research uses a high-performance vector-drawing refresh-tube display, such as those made by Evans and Sutherland and by Vector General. These are supported by high-powered minicomputers, with 64K-256K bytes of core, a disk, and a complement of keyboards, tablets, and joysticks. A (broad) link to a powerful time-sharing computer with a full file system is very valuable. We currently operate a Vector-General display on an 88 Kbyte PDP 11/45, attached to a 2 Mbyte System/360 Model 75.

The second requirement is software support -- a suitable operating system, a graphics subroutine package, inter-machine communications, and a language compiler. Our software system includes a DEC operating system, a PDP-11 PL/C compiler built on the System/360 PL/C compiler from Cornell, and many components built by students under the leadership of James D. Foley and P. J. Kilpatrick.

The most important requirement is commitment on the part of the ICG investigators. Building any serious ICG application is a lot of work. Each requires thousands of program statements, which must be designed, written, debugged, and documented. I have visited university and industrial computation centers where expensive ICG equipment sits idle. Inquiry generally reveals that the institution bought the equipment, hoping investigators would use it. This is the wrong order. Until deeply committed investigators are clamoring for the equipment, its acquisition is a waste.

The fourth requirement is for users, who have real applications that might potentially benefit from an ICG attack. These, however, if other than the investigators, must not have their hopes aroused too soon or too keenly. Development will be long and halting; too many promises bring disillusionment and discontent.

2.3 Realism and abstraction

Most of the work done to date on ICG consists of the development of applications. Most ICG research, as opposed to development, has addressed techniques for forming images, for structuring image and model data, and for expressing human manipulations. In a pair of brilliant papers delivered at the 1965 IFIP Congress, Ivan E. Sutherland gave a unifying concept to the existing work in ICG and laid out a program or research on that concept, [3] and [4].* He considers the display screen as a window, through which the user looks at a virtual, conceptual 3-D universe. The display devices and programs allow one to select what is to be seen and from what viewpoint. The underlying computer program maintains the virtual universe and all its interrelationships, as the user changes parts of it from the console.

Sutherland's research program called for inventing ways to make the image in the window more and more realistic, until at last it becomes indistinguishable from the image in a real window, a real window augmented with magical powers of scaling, cross-sectioning, labeling, etc.

To a surprising degree, not only Sutherland's own research, but most ICG techniques research, has been carrying out Sutherland's program, and an impressive amount of progress has been made.

But realism is not enough. Were Sutherland's realism program completed perfectly, it would meet the needs of only part of the applications of ICG, those involving visible objects.

For a second class of applications, the power of ICG lies purely in the representation of abstractions. This is true of all electrical phenomena, of molecular phenomena, of data graphing and curve fitting, of network and graph theory.

For a third class of applications, the power of ICG comes from the ability to superimpose abstract and realistic representations, as in cartography.

For the study of abstractions, the work on visual realism as such is irrelevant. So we plot stick figures of molecules, straight-line schematics of electrical circuits, graphs of voltage versus time, etc. Through familiarity these abstractions become very real. Most electrical engineers will answer without hesitation the question, "What does an alternating current look like?"

3. ESTABLISHING THE POWER OF INTERACTIVE COMPUTER GRAPHICS

Since the days of Whirlwind I, computer scientists have believed that ICG significantly improved communication with computer users. But the justification for using ICG has been long on intuition and short on quantitative analysis. There is very little proof of effectiveness in the ICG literature.

The first researches in our laboratory undertook to determine whether this faith could be supported with fact. Can this new tool give new power?

3.1 Oliver's experiment

P. Oliver developed an ICG system for teaching

---

*The summaries in the Proceedings are woefully incomplete, but the ideas given then are clearly set forth in Sutherland's 1970 paper. [5]

selected methods in numerical analysis. [6] After it was complete, he tested it with two non-credit courses: an experimental group and a control group.

The control group was conventionally taught. The experimental group was taught by a lecture on the theory of the method in question, illustrative examples presented by the instructor with the display, and review. Both groups did similar lab exercises; the experimental group did theirs with the display.

Oliver's experiment was of non-randomized, pretest-posttest design. The sample sizes were small and hence the statistical tests were weak. Nevertheless, the use of the interactive graphical system significantly improved performance on three of four tests given, and apparently improved it on the fourth.

Many of Oliver's valuable qualitative observations, though made in a teaching situation, we see to apply to many other ICG situations as well. These include:

1. Building even this limited application software took a lot of effort, some 1200 man-hours.

2. Computer systems fail at times; more complicated ones fail more often. The user must have a planned mode of backup operation.

3. The most fruitful part seems to have been the hands-on time by the students. Individual manipulation of the mathematical objects improved perception and understanding of them.

4. Students taught with ICG had much greater class participation, were eager to pursue associated topics, showed initiative in using the system in unexpected ways, and developed more systematic ways of approaching problems.

3.2 Prokop's experiment

J. S. Prokop studied the power of ICG on decision-making in a business environment. [7] He gave a course in modern techniques of inventory management to 22 practicing managers of inventories, recruited from nearby industry. At the end, each manager worked through two 24-month simulations. In each, a set of twelve different management policies, embodying various combinations of forecasting, smoothing, and reordering methods, was followed against a single simulated demand. Each (simulated) month the manager was presented with the results of following all twelve policies. Each month he was asked to rank the twelve from best to worst, and he was asked to make a "final" ranking recommendation, as if to his own boss, as soon as he felt comfortable doing so. At any time, the manager could review the results of any or all previous months.

A randomized Latin square design was used, with each subject being in the experimental group on one simulation, the control group on the other. The Latin square made it possible to separate inter-group differences and order effects from differences due to the use of graphics.

Both experimental and control groups had the results of the month-by-month simulations available to them in tabular form and in plotted form. The experimental groups had them at their fingertips via ICG; the control groups had the same results printed or plotted on paper. In simplest terms, the ICG system was a random-access electronic page turner.

Prokop measured three variables:

- clock time required to make decisions,
- simulated month (1 to 24) in which a "recommended" ranking was made,
- quality of "recommended" ranking, as measured by consistency between it and the ranking actually made by the same subject after seeing all 24 months' results.

Note that Prokop did not attempt to evaluate the rankings themselves. This is a value judgement, based on balancing lost sales against inventory carrying costs, and he wanted each subject to apply the judgement he had acquired from experience. Therefore consistency between the ranking with imperfect information and that with perfect hindsight was used as the quality measure.

Prokop's statistical results are very convincing:

1. Mean clock time to decision was

graphics	52 minutes
paper	82 minutes

a difference statistically significant at the 0.1% level.

2. Mean month to decision was

graphics	9.2 months
paper	11.3 months

a difference significant at the 1% level.

3. Individuals tended to make <u>better</u> decisions with the graphics output, even though they made them on two months' less data. This difference was significant at the 5% level.

The first result is to be expected; the mechanics are simpler. The third result was surprising to us -- we should have been happy had the decisions merely turned out to have been no worse, though earlier, when graphics were used.

The second result is most important economically, and the most exciting to the toolsmith. The businessmen armed with ICG display of this market data <u>understood</u> the trends in it two months earlier than their competitors who had the same data, tables, and plots on paper. Who would have expected so much from an electronic page-turner?

Oliver's and Prokop's experiments definitely confirmed quantitatively our intuitive belief that ICG tools of considerable power can be built. Wherein does the power lie?

1. As is well known, great power comes from <u>visualization</u>, that is, from graphics whether computer-produced or not. A picture is worth a thousand words. This effect is clearly present in Oliver's results, but not in Prokop's, where experimental and control groups both had pictures.

2. There is also known to be power in <u>responsiveness</u>, in interaction, whether graphic or not. The start-up time for a train of thought is fairly long; any interactive system whose response is fast enough does not tempt one to interrupt his thought train, and higher productivity results. I think this effect is essential to explain Prokop's results. Moreover, such systems <u>command</u> attention. This is very evident in Oliver's observations of not only laboratory behavior but even of classroom behavior, where the students only watched him use the graphics system.

3. To these established sources of the power of ICG tools, I would add a third that depends upon the combination of graphics and interaction.

   Both Oliver's and Prokop's results point to the perception-enhancing effects of <u>manipulation</u>. That is, manipulation in an ICG system is not only useful for facilitating changes in the model being depicted, it also improves understanding of the model. Moving images on a screen have great power to inform; images that move <u>in response to</u> one's manual manipulation seem to be perceived more as real things, and studied more intently. Such systems achieve a much higher degree of <u>transparency</u>. We observe that the hands-on user quickly becomes unconscious of the screen and the system and concentrates on the underlying model whose attributes are being depicted.

4. TOOLS USING MORE SENSES

If the watching of imaginary objects move and alter as one manipulates them endows them with a kind of real existence, and if this process yields more power for understanding and designing imaginary objects, can we build yet more powerful tools by using more senses?

Indeed so. In aircraft piloting simulations, for example, the visual display is supplemented by an audible display simulating motor and wind noise to furnish cues as to throttle setting and airspeed. Such audible cues are very effective supplements to the normal cockpit visual indicators. The effectiveness of the audible output arises partly from the fact that it uses a separate input channel to the man, providing a continuous cue without competition from other visual tasks, and partly from the fact that the signal is assimilated and acted upon subconsciously.

In 1965, Sutherland suggested that similar benefits could be derived by using the human kinesthetic sense as yet another independent channel to the brain, a channel whose information is assimilated quite subconsciously. He suggested that this might be done by means of a force-feedback electrical remote manipulator, such as is used in handling radioactive materials. [3]

The operator of such a manipulator experiences the illusion of grasping objects, encountering the table top, etc., even though his master manipulator and the slave manipulator are connected only electrically. Motors in the master re-create and transmit to him the forces experienced by the slave manipulator. [8]

The feedback to the user of such a device is composed of sensations to the propriopositional system (where in space one's limbs are) and force feedback, which is sensed as a pressure. The tactile sense as such is not used. Since the operator is always in contact with the finger-guides of the master hand, he does not get tactile information when the slave hand makes a contact. He only gets force feedback when he tries to move the master by applying pressure to it, and he finds that the resistance to motion has sharply increased. The experience is rather like exploring a space with the point of a screwdriver that is tightly gripped by the hand.

Sutherland suggested that one substitute a computer for the slave station. Then the operator would use the master as though he were performing a task in real space via the remote slave, but the computer would carry out the task and return the proper feedback signals according to events in the conceptual space. By such means one can explore, and <u>assimilate the results of exploring</u> force fields that are microscopic, astronomical, or otherwise not reproducible on earth.

Our laboratory has investigated such tools in two experiments, both dealing with simulation of real objects. We hope to extend the work to tools operating upon abstract objects, e.g., models of chemical molecules. Other work using the kinesthetic sense has been reported by A. M. Noll [9] and K. R. Wilson.

4.1 <u>Batter's 2-D experiment</u>

Before launching out into a full-scale remote manipulator system, which involves seven degrees of freedom (three at the shoulder-elbow, three at the wrist, and tong pinch), we decided to test the concept on a simple two-dimensional pilot system. J. J. Batter and I reported this work at the 1971 IFIP Conference. [10] I shall briefly review.

The manipulator/force display was a small knob, attached to a movable platform that could be positioned anywhere within a horizontal plane two inches square. Potentiometers sensed its x and y positions; servomotors exerted x and y forces on the knob. Both were connected to the computer driving an associated visual display. The software completes an ICG system that enables the user to examine elementary force fields by experiencing a force proportional to that that would be exerted on a particle in the field.

As the user moves the knob on the force display device, a visual cursor follows his motion. At the same time, he experiences a force. The magnitude and direction of this force is also indicated on the screen by a vector originating at the position of the particle. The force and visual displays are recalculated and updated approximately 12 times per second, depending upon the complexity of the force calculation.

Batter undertook to measure the effectiveness of such a tool for improving learning the properties of force fields. [11] He used subjects who had not studied force fields. The participants were paid according to their performance.

The subjects were randomly divided into test and control groups. The whole class, both groups, were given six hours of lectures. Then they were given a pretest examination.

Each student was asked to "map" five fields. He was given a sheet of paper with the center of the field indicated, the type of field to be mapped, and a sample vector to indicate scale. Ten numbered dots indicated positions in the field and the students were asked to estimate the magnitude and direction of the force by drawing vectors at these ten points.

After the pretest, each student examined some 16 force fields in two hours of exercises. The members of the test group received force feedback while members of the control group did not. This was accomplished by unplugging the servo motors for the control group. This ensured that all visual displays, timings, etc., were identical for the two groups. When all the students had completed the exercises, another examination was given, identical in type and scoring to the pretest.

The experiment was repeated three times. The subjects for Group A were science majors from an honors section. This group's learning was better than the control group's, a difference significant at the 2.5% level.

Subjects for Group B were students from a physics section for non-science-majors. The results for Group B were quite different. Only slight improvements were noticed, explainable by chance. Puzzled, we selected a third group, once again from a physics-major section. The results replicated those of Group A.

The qualitative observations offered a clue. The science-oriented students showed greater interest in the material presented and in using the device. The non-science students tended to watch the clock frequently. They spent more time examining the familiar inverse square field and the three-body field than any others. The science students became deeply involved in the use of the device, whether force feedback was present or not. They seemed oblivious to other activities in the room and their attention could be attracted only with difficulty. Six of these students talked to themselves.

The lesson? A tool can only be useful when the user wants to use it. Our monetary incentives were insufficient to motivate Group B. Groups A and C found interest and inherent motivation in the experiment, with or without force feedback.

Frankly, I continue to be surprised and puzzled by this result. I had expected the display and manipulation to add enjoyment and motivation, closing the gap between those of low and high intrinsic motivation. Instead, to those who had much, more was given by the more powerful tool. Those who had little were less helped. To my knowledge, none of the published studies of computer-assisted instruction treat this question.

4.2 Kilpatrick's 7-D experiment

Last year, P. J. Kilpatrick completed building and evaluating a full-scale remote-manipulator kinesthetic display, GROPE-II. [12] He tested it by subjects grasping and manipulating imaginary blocks on and over the surface of an imaginary table. As in Batter's experiment, the force feedback could be used or disabled without changing any other part of the display or environment. The full discussion will be published elsewhere; Fig. 1 shows the arrangement. I here summarize the most important results.

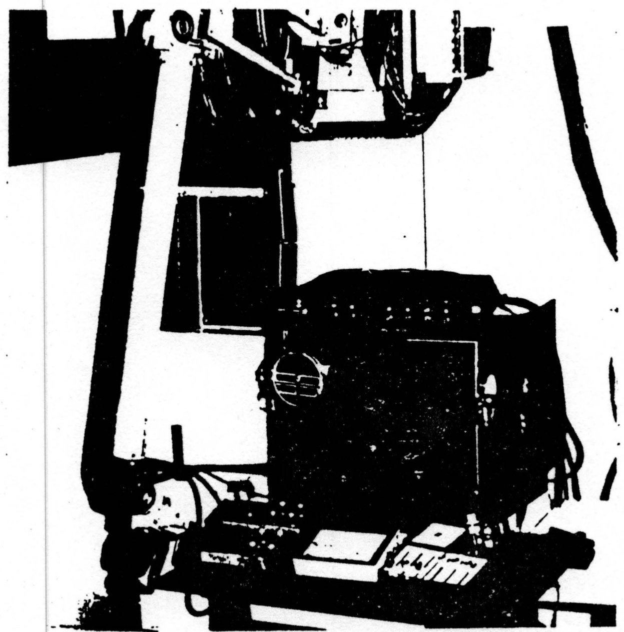

Fig. 1. Kilpatrick's experimental kinesthetic display.

1. A kinesthetic display used as an adjunct to a visual display enhances both perception and the performance of simple motor tasks.

2. A set of realism-enhancing cues, each of which individually improves manipulative performance, were compared against each other pair-wise. These included several monocular depth cues, true stereoscopic display, and force feedback.

   The force cue proved to add more to manipulative performance than stereoscopic display, more than variable viewpoint, and less than more powerful monocular position cues.

3. Posting of new data to sample-and-hold force display must be done more than 15, and not necessarily more than 20 times per second if the illusion of continuous force change is to be achieved and if hard-surface illusions are to be satisfactory. Batter found 12 postings per second to suffice when hard-surface illusions are not necessary.

4. Users of an imperfect-perception visual system, whether ICG or closed-circuit television, tend to decompose multi-dimension positioning tasks into several separate subtasks, each of lower dimensionality. This is in contrast to normal eye-hand coordination behavior.

   Thus, a subject required to move a point probe to a target in 3-space moves smoothly and continuously if he is really seeing the probe and the target. With imperfect perception, our subjects unanimously tended to position in two dimensions as one task, and then position in the third, or *vice versa*. When a three-degree-of-freedom orientation was also part of the task, this too was decomposed into a 2-D fit and a 1-D fit. Thus a

6-D fitting task was usually decomposed into four tasks, each 1-D or 2-D.

5. Different users decompose n-dimensional fitting tasks in different ways and use both visual and force cues in different ways. How useful any particular cue is to a particular user is a complex function of his problem attack. This suggests that any real system should either (a) be custom-tailored to each of its intended users, or (b) have a redundant set of cues, so that a variety of users can each adopt satisfying strategies.

6. A kinesthetic display has the unusual property that the same physical device is used for input and output. Therefore the presence of a force output can substantially reduce the user's speed and precision in generating position and force inputs. Users are typically quite surprised by the interaction.

   The problems raised by this interaction may be very fundamental. Weber's work [13] warns us that the perceptual kinesthetic space is non-Euclidean and is systematically distorted by the presence of loads on the sensors. Moreover, the distortion is not only non-linear, it is non-monotonic, hence not readily linearized.

7. An analog of Gregory's "object hypothesis" model of visual perception [14] appears to hold for force perception. Gregory postulates that when one views an object or a picture of an object, he forms a hypothesis about its structure which he uses to predict how it will look from other viewpoints. He believes one tends to select the most probable structure to hypothesize, making it "difficult, perhaps sometimes impossible, to see very unusual objects."

   Kilpatrick postulates, based on his observations and Batter's, that displayed forces are similarly interpreted as the most likely forces. Since we commonly experience only constant or linear force fields correctly. Batter found, for example, that subjects were unable to distinguish among square-law, cube-law, and other high-power force fields without direct comparison.

   A conclusion is that users of a kinesthetic display system must be trained in the perception of unfamiliar forces before such a tool will be useful.

8. Experiments from the psychological literature show that when concurrent visual and kinesthetic cues conflict, the visual ones dominate. Kilpatrick was able to use this effect to give surprisingly satisfactory simulation of hard surfaces with what was in fact a relatively soft Hooke's Law behavior. I believe the same effect accounts for the fact that we observed no difficulties due to Weber distortion of the kinesthetic space: we provided visual observation of a Euclidean space, and that perception dominated.

   For this and other reasons, we believe kinesthetic displays will be useful tools chiefly, and perhaps only, as adjuncts to visual displays.

9. Enough is known in the psychological and ergonomic literature to allow one to spell out the desirable kinesthetic, anthropometric, perceptual, and engineering characteristics of kinesthetic displays; Kilpatrick has detailed these.

10. Not only is kinesthetic display hardware somewhat expensive, the costs of application and systems software are substantial. Kilpatrick's GROPE-II system consists of some 12,500 PL/I and assembly language source statements. Many others were written but not finally used. Approximately 5.5 man-years went into the software. Considerable computer power is required to maintain real-time interaction.

5. LESSONS FROM A REAL TOOL -- A GRAPHICS SYSTEM FOR MACROMOLECULES

In order to keep our feet on the ground, we have from the beginning of our ICG research been growing an application system intended to help real, independent users solve real problems. This application work has continually affected and benefitted from our concurrent work on techniques.

All of this work has centered on the display and manipulation of macromolecules -- proteins and nucleic acid structures. These structures are central to the study of biochemistry. They contain from hundreds to tens of thousands of atoms, are too complex for purely computational attacks to be successful, and yet demand computation for many aspects of their study. Their three-dimensional conformations are crucial to their biochemical actions. Visualization of their shapes is essential for understanding them. They are readily represented by graphs of straight-line segments. For all these reasons, an ICG tool promised to be buildable and usable.

The first such tool, a pilot model, was built and demonstrated by C. Levinthal at MIT in 1966. [15] Since then about twelve teams in the U.S. and Europe have built such tools. I shall not here survey the similarities, differences, and special contributions of these several systems. It will suffice here to review the scientific results and engineering lessons we have learned from our experience.

5.1 Wright's GRIP-71 system

W. V. Wright, working with biochemist J. Hermans as his principal client/advising chemist, developed in 1971 a system designed for the display and study of proteins. [16] Hermans is especially interested in the energetics of protein conformation and conformation change, so the system allowed one to see stick-figure models of proteins, to move the atoms relative to each other, and to calculate, interactively, the internal energies of molecules and assemblages, and the force and torque vectors acting on portions of molecules.

The principal results and lessons from this system were:

1. The system, even in its pilot implementation, was useful to the client, proved much more flexible than physical models, facilitated force and energy calculations, and provided a tool capable of use on real problems on the frontier of molecular biology.

2. A knowledgeable and patient client is the most

valuable asset a toolsmith can have. Hermans and his associates spent 16 sessions of 90 minutes each using the system while Wright watched and suggested techniques. Hermans' concept of what he wanted in such a system naturally evolved during this experimental phase; Wright found this intense observation of users at work to be the richest source of explicitly determined needs. We believe such a tool cannot be properly constructed without intensive and extensive trial use on real problems.

3. A toolsmith-user pair has several advantages over a combined user-toolsmith person. First, the user is not distracted by tool-implementation issues and can keep his attention on the research problem. This focus on the end problem helps the tool-building. Second, the implementation of such a tool involves rather sophisticated data representations and operations, and the specialized technical knowledge of the computer scientist is very useful. GRIP-71, for example, offers the user two levels of specialized virtual computer, one for molecule manipulation, and an inner one specialized for abstract 3-D geometry. Third, the toolsmith functions as observer in each session to identify the user's real hierarchy of goals and to see how these might be more easily reached.

4. Such a tool must be so useful that a senior researcher will want to use it himself; it must be so easy to use that even a full professor can use it. If users send their graduate students instead of coming themselves, that is a symptom of failure: either the power or the accessibility of the tool is lacking.

5. The combination of power and ease of use requires specialization of the user language, whether it be alphanumeric, command-button, or analog in form. We found the molecular application to require almost 100 distinct command verbs.

6. These 100 commands turn out to cluster nicely into relatively small subsets for each conceptual work-unit. That is, there is a high degree of locality within observed command sequences. One can thus implement a set of selectable menus, and menu-switching happens only a few times per hour.

7. An inherent conflict exists between power, which calls for many commands, and ease of use, which calls for few. This can be addressed by providing an extensible command language in which new commands can readily be defined; starting with a small set; and building as use dictates.

8. The commands needed fell into only two levels: primitives, and macro commands defined as sequences of primitives. Nesting of macro commands was easy in GRIP-71, but it was used only rarely.

9. Primitive commands must be selected by an outside-in process, as sample work-units are analyzed into component parts. In GRIP-71, these proved to be chiefly 3-D geometry operators on points and vectors.

10. Macro commands, on the other hand, are selected by an inside-out process as frequent sequences are observed in actual use.

11. With large command sets, ease of use can be facilitated with <u>prompting messages</u>. Effective prompting is most easily achieved when the command language uses Polish notation (operator, operands), as opposed to algebraic (operand, operator, operand) or reverse-Polish (operands, operator). Moreover, Polish notation lends itself to hierarchically structured menus. On the other hand, reverse-Polish lends itself to command chaining, also a powerful aid.

12. Of the criteria laid down by Foley and Wallace [17], we observe tactile continuity and visual continuity to be major factors in the ergonomic design of ICG systems. Wright observed the need for a touch keyboard for GRIP-71 commands, to maintain continuity of screen watching.

13. User satisfaction with system response times does not depend upon any absolute threshold but, for each command separately, upon the ratio of actual response-time to a mental model of its computational work. For trivial commands an average response of even 0.85 seconds was observed to be irritating. For an internal energy calculation, 45 seconds is quite acceptable. Frequently used trivial commands should have streamlined implementations.

14. The molecular graphic system is useful not only for the study of molecules, but also for the auxiliary purposes of (a) communication among researchers and with students, (b) preparation of illustrations for publication, and (c) immediate, startling perception of errors in input data.

15. The chief observed inadequacies of the GRIP-71 system for its intended purpose all appeared to be related to its lack of dynamic motion:

(a) Insufficient cues for good 3-D perception,

(b) Even though the system used a button language instead of a character-string language for commands, <u>any</u> discrete language is poor for manipulation of objects in 3-space. One wants continuous analog input,

(c) One always wants a direct, analog way of changing the viewpoint for viewing any 3-D object.

16. The use of perspective transformations hinders, rather than helps, in the 3-D perception and fitting of molecules. I do not know why: whether because we are working with highly abstracted objects, or whether because perception of parallelism is especially important in molecular structures.

17. An important goal for such a system, not seen when we started, is to facilitate <u>methodological invention</u> by the user, and to capture it when it occurs.

This means one needs the power to implement new verbs, views, mathematical evaluations, etc. quickly, while the client is still excited about trying them, and ergonomically smoothly, so they don't distract. Efficiency in computer use or memory use is irrelevant at this stage. Those new techniques which work can be reimplemented efficiently at a more normal pace.

5.2 The GRIP-75 system

After GRIP-71, our exploratory system, we undertook to build a more powerful pilot tool, capable of production use. The GRIP-75 system has been designed and built by E. G. Britton, J. S. Lipscomb, and M. E. Pique, who with many others worked under W. W. Wright's direction. [18]

GRIP-75 was designed to have more specific application than GRIP-71. Britton, working with Duke biochemist Sung-Hou Kim as his principal client/advising chemist, defined the system to be useful in fitting molecular models to electron density maps produced by X-ray crystallography. The major new facilities were

(a) ability to contour and display electron density maps, with molecules superimposed,

(b) direct analog input of viewpoint, of viewing controls, and of translation and rotation manipulations of molecule pieces by joysticks and knobs.

(c) incorporation of an on-line mathematical routine relaxing distorted molecules toward idealized bond lengths and angles. This relaxation program was derived from a more elaborate one by J. Hermans and J. E. McQueen.

The system was placed into productive use in July, 1975. Since then, Lipscomb, working under V. L. Wallace, has completed a comprehensive set of 3-D perception aids. [19] Pique, also working under Wallace, has substantially supplemented the direct translation and rotation of molecule pieces by concurrently permitting multiple simultaneous rotations about twistable bonds, with the rotations each directly specified by an analog knob. [20] Kim, and Duke biochemists D. and J. Richardson, have been major clients giving much important advice.

The principal results and lessons from this system are:

Effectiveness

1. The system, although intended as a pilot system, has been conclusively demonstrated to be effective. It has been used by nine different teams of biochemists, from six universities, for a total of about 1000 system hours.

2. At least seven research papers reporting biochemical results achieved in part with its aid have been published.

3. At least one protein molecule, sea-snake neurotoxin, has been fit to its density *ab initio*, without a physical model with atomic detail being built; this is a first.

4. Users report speed-up ratios of four times to ten times for the fitting process.

System lessons

5. Much of the effectiveness comes from the total-system analysis of the crystallographic process done by Britton with help from biochemists J. L. Sussman and J. E. McQueen. This process requires computation of electron densities, contouring of density maps, fitting proper, and mathematical refinement. Integrating the fitting system with the big computer allows data to flow smoothly between batch and interactive functions, and the contouring and idealization usually done in batch can be brought on-line.

6. The system features most warmly liked by users include those that depend upon access to the big fast computer, its big memory, and its big files.

7. The back-up and recovery provisions are crucial to user satisfaction.

8. The total system available to the new biochemist user includes a computer science graduate student who serves as shepherd, explaining more obscure features, preparing data, suggesting means towards goals. This person is an important part of the tool.

9. Tool-building for a clientele rather than a single client frees the toolsmith to seek more general solutions to users' problems.

Software lessons

10. This experience has demonstrated to us, once again, the lessons on productivity, on structuring, on the virtues of high-level language, and on documentation that I previously learned on a larger project. [21]

11. The building of nested language levels, with Wright's geometry language between the PL/I level and the user manipulation level has enabled us to give quick response to several user requests for new function.

Ergonomic lessons

12. The use of analog joysticks and knobs to express manipulations has made the largest single improvement in the ease of system use. This power is not without cost. The continual reading of these and the posting of their changes uses up most of the computing speed of the PDP 11/45.

13. The transfer of most variables to analog devices has brought a new problem in tactile continuity -- one needs three or four hands, and moving from device to device is distracting. I think our next system will reintegrate all 1-D and 2-D controls onto analog inputs via a single data tablet, keeping the 3-D controls as joysticks.

14. The environment becomes very important when users work 4-8 hour continuous sessions. Room noise, temperature, and illumination become essential parts of system design.

15. Workstation controls must be very thoughtfully

positioned. Videotapes of user sessions must be studied and restudied to see awkwardnesses.

16. We observe these users to make 3-D and 6-D fitting manipulations by dimensional decomposition, just as Kilpatrick's subjects did. This suggests that some radical simplifications of our controls might be possible.

17. As with Kilpatrick's subjects, GRIP-75 users adopt a wide variety of approaches to what seem to be very similar tasks. Even in viewing, some prefer stereoscopic viewing; others prefer more elaborate contouring. Some prefer to manipulate by bond-twisting; others prefer to cut a subassemblage off of the molecule, move it to a good fit, and then repair the bond length and angle distortions so introduced.

    An essential part of good design of such systems seems to be the provision of a redundant set, even a rich set, of viewing and manipulating techniques. Each user then chooses the working vocabulary that best allows him to think about his problem, rather than about our tool.

6. DESIGNING WELL-FITTING MAN-MACHINE SYSTEMS

The architect Christopher Alexander, in his Notes on the Synthesis of Form [22], makes the penetrating observation that the only way to achieve good fit between any design and its requirements is to find misfits and remove them; there is no direct way to derive form from requirement. Good fit is the absence of all possible misfits. This he supports with convincing arguments.

This I find to be an overarching lesson from all our graphic system design work. We observe that we have not found a direct design procedure for the man-machine interface; Alexander shows that we never shall. Principles we have found; we shall find more; and these will guide design. Satisfactory man-machine systems, however, will always be the product of iterative design in which misfits are painstakingly removed. I think the only effective design methodologies will be those built around this iterative approach.

REFERENCES

[1] H. Zemanek, Was ist informatik?, Elektronische Rechenanlagen, vol. 13, no. 4, August 1971, 157-161.

[2] John 5:44.

[3] I. E. Sutherland, The ultimate display, Proc. IFIP 65, vol. 2, 506-508.

[4] I. E. Sutherland, Three kinds of graphic data processing, Proc. IFIP 65, 582-583.

[5] I. E. Sutherland, Computer displays, Scientific American, vol. 222, no. 6, June 1970, 56-81.

[6] P. Oliver and F. P. Brooks, Jr., Evaluation of an interactive display system for teaching numerical analysis, Proc. AFIPS FJCC, vol. 35, Fall 1969, 525-633.

[7] J. S. Prokop and F. P. Brooks, Jr., Decision making with computer graphics in an inventory control environment, Proc. AFIPS FJCC, vol. 37, Fall 1970, 599-607.

[8] R. Goertz, Manipulator systems development at Argonne National Laboratory, Proc. 12th Conf. on Remote Systems Tech., 1964, 117-136.

[9] A. M. Noll, Man-machine tactile communication, Ph.D. Dissertation, Polytechnic Institute of Brooklyn, Brooklyn, New York, 1971.

[10] J. J. Batter and F. P. Brooks, Jr., GROPE-I: A computer display to the sense of feel, Proc. IFIP 1972, 759-763.

[11] J. J. Batter, The GROPE-I system: A user's manual, unpublished research report, University of North Carolina, Chapel Hill, North Carolina, 1975.

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

[13] C. O. Weber, The properties of space and time in kinaesthetic fields of force, American Journal of Psychology, vol. 38, no. 4, 1927, 597-606.

[14] R. L. Gregory, Eye and brain, 2nd edition, McGraw-Hill, New York, 1973.

[15] C. Levinthal, Molecular model-building by computer, Scientific American, vol. 214, no. 6, June 1966, 42-52.

[16] W. V. Wright, An interactive computer graphic system for molecular studies, Ph.D. Dissertation, University of North Carolina, Chapel Hill, North Carolina, 1972.

[17] J. D. Foley and V. L. Wallace, The art of natural man-machine conversation, Proc. IEEE, vol. 62, no. 4, 1974, 462-471.

[18] E. G. Britton, A methodology for ergonomic design of interactive computer graphics systems, as applied to crystallography, Ph.D. Dissertation, University of North Carolina, 77 Chapel Hill, North Carolina, 1977.

[19] J. S. Lipscomb, Three-dimensional display of molecular models, M.S. thesis in progress, University of North Carolina, Chapel Hill.

[20] M. E. Pique, Nested dynamic rotations for molecular graphics, M.S. thesis in progress, University of North Carolina, Chapel Hill.

[21] F. P. Brooks, Jr., The mythical man-month -- Essays on software engineering, Addison-Wesley, Reading, Massachusetts, 1975.

[22] C. Alexander, Notes on the synthesis of form, Harvard University Press, Cambridge, Massachusetts, 1964, Chapter 2.