D E S I G N
A N D THE
C OMP UT ER
D E
S I
G N
A ND T HE
COM P U TE R

DESIGN AND THE COMPUTER

Design Quarterly

66/67

DESIGN QUARTERLY 66/67
Editor: Peter Seitz
Circulation: Gale Sharpe

Design Quarterly is indexed in Art Index
Subscription rates are 4 issues $3.50, 8 issues $6.25, 12 issues $9.00
Single issues $1, Double issues $2
Foreign postage $1.00 for 4 issues. Design Quarterly is published by
Walker Art Center, 1710 Lyndale Avenue South, Minneapolis, Minnesota 55403

Change of address: To insure receiving all copies, give the old address as
well as the new one and allow five weeks for change to become effective.

WALKER ART CENTER
1710 Lyndale Avenue South
Minneapolis, Minnesota 55403
Martin Friedman, Director

T. B. WALKER FOUNDATION

Trustees
Archie D. Walker, President
Walter W. Walker, Vice-President
Hudson D. Walker, Vice-President
Dana C. Smith, Vice-President
Justin V. Smith, Secretary and Treasurer
Mrs. Malcolm A. McCannel, Assistant Secretary
and Assistant Treasurer
Theodore S. Walker, Assistant Secretary
Mrs. Theodore S. Walker
Mrs. Willis J. Walker
Mrs. James van Loben Sels
Brooks Walker
Mrs. Wellington Henderson

Honorary Trustee
H. Harvard Arnason

Members
Mrs. Calvin Yeates
Mrs. Calvin Goodrich

WALKER ART CENTER BOARD OF DIRECTORS

David M. Winton, President
Pierce Butler, First Vice-President
Mrs. Edmond R. Ruben, Second Vice-President
Martin Friedman, Secretary
Donald C. Borrman, Treasurer
Mrs. C. Merritt Case, Jr.
John Cowles, Jr.
Mrs. Kenneth Dayton
Norton Hintz
Donald W. Judkins
Mrs. Malcolm A. McCannel
Mrs. Richardson B. Okie
David H. Preus
William G. Shepherd
Justin V. Smith
Philip Von Blon
Archie D. Walker
Walter W. Walker
Fred Weil, Jr.
Mrs. O. Meredith Wilson
Louis N. Zelle

Ex-Officio
Hon. Arthur Naftalin
Stuart W. Rider, Jr.
Bruce Smith

Copyright 1966 Walker Art Center

Design Quarterly 66/67

Design and the Computer

Walker Art Center

Minneapolis, Minnesota

CONTENTS

INTRODUCTION

Continuous technological advances have put a growing demand on the design professions. Recognizing the need of information for the designer, DESIGN QUARTERLY presents in this double issue, "Design and the Computer," some of the more important recent developments in the application of computer technology to design tasks. In order to answer fundamental questions such as: *"Why is the computer needed in the design process?"* and *"How is the computer used?,"* this issue necessarily became more technical than usual, although all the essays presented deal with the application of the computer to daily design problems.

The survival of the design professions and, in fact, of our whole society depends upon the acknowledgment and study of new conditions and technologies. The designer needs to formulate a new design perspective which includes increased social responsibility and a recognition of technological achievement in order to deal with the problems presented by a changing society. The role and function of the architect, graphic designer, city planner and industrial designer has grown increasingly complex and responsible: the design for better learning, living and communication.

The concept of "good design" achieved by a combination of intuition, intelligence and good taste, which is still the primary concern of today's designers, is no longer adequate. Too many "tasteful" variations of all types of objects exist, but what we need most are better cities, mass transportation systems, hospitals and highways. The traditional ways of thinking, perception and communication have become obsolete and none of these complex problems can be solved by experience and intuition alone. Most design techniques are outmoded too, because they merely cope with the factors and conditions resulting from the Industrial Revolution and are not yet concerned with the electronic age and its implications. Design must move into the area of electronic data processing because new methods and tools are needed to gather the mass of data and extend the comprehension of today's major design problems.

The computer is the tool which the designer will have to learn to use. Most designers are unaware of the existence of such a tool and only a few architects, graphic designers and industrial designers in different localities, corporations or institutions are beginning to explore the potential of the new computer technology.

In solving design problems, the designer relies mainly on experimentation with ideas. Concepts are formed and changed, associations occur, are examined, and then approved or rejected. Once the basic idea or structure is formulated, most of the ensuing work is mechanical. The computer can free the designer from many of these tedious and mechanical tasks. For example, it can generate a perspective view of an object which can be drawn by a plotter (automatic drawing equipment), present the object on a television-like display tube in different perspective views, or rotate the displayed object in space just as if the designer were holding it in the palm of his hand. Such possibilities enable the designer to examine a solution, modify it or start anew before it is ever executed in concrete or steel. If a three-dimensional model is required, the computer can be programmed to produce a tape which controls a milling machine to carve out the model in metal.

The computer not only assists the designer in his manual tasks, but because of the tremendous command of information made possible by the computer, it also enables him to process almost endless facts and figures, organize this complex data and analyze the requirements and interactions of any design problem. Beyond this, the computer may also be used as a creative medium to produce drawings, electronic sculpture, choreography and movies.

During the past few years, terms such as *computer hardware* and *software* have evolved. These terms are related to the workings of the computer: the information needed to perform a task and the procedures to be followed constitute the program which is usually entered into the machine on a typewriter-keyboard. The program is translated into electronic impulses and often stored on a magnetic tape. The control unit of the computer performs the work specified by the program and the resulting information is transferred to an output device that prints messages on pages, punches cards or tapes, or displays images on a television-like screen.

These input and output devices, arithmetic and control circuits, and a memory unit, are the so-called computer *hardware.* The programs of instructions and various machine languages which put the computer to work are the *software.* The question of how the designer, unfamiliar with computer technology, can use the computer, fortunately seems to become less and less important. Programming languages in written and even spoken English are possible in the future, and already today the designer can communicate visually with the computer. In the future he may be able to use the computer with no more knowledge of the operations performed than the average person knows about the workings of a telephone.

Computer technology is a rather recent invention if we consider that the first commercial computer was marketed as recently as 1950. In the past ten years, the typical electronic data processor has become 10 times smaller, 100 times faster and 1000 times less expensive to operate. With the development of multiprogramming and the time-sharing system where one user shares a computer with several hundred others, with input devices located in offices, laboratories and private homes, the computer can serve as a community pool of knowledge and skill from which anyone can draw according to his needs.

The list of areas in which the computer has been in use for some time is impressive: marketing, banking, management, steel production, assembly lines, law enforcement, statistics, poll and predictions, operation of ships, railroads and airplanes, language translation, power plants, teaching, map making, tax returns, medicine, economic planning, space exploration, library science, military planning and scientific experimentation, etc. More important for the designer, however, are the following areas in which the design has been facilitated by the computer or where the potential of computer-aided design is indicated by experiments: civil engineering, transportation planning, engineering design, architecture and urban planning, automotive design, publishing and printing, graphic design, film making and art.

For example, in *engineering,* computer programs aid in designing bridges. The engineer-designer can make free-hand drawings of a bridge on the computer system's television-like screen, the computer will convert the rough design into exact engineering specifications, calculate and display stress, and show the design in whole or in part, or in any perspective. This includes the possibility of changing the design or modifying parts of it, with the computer incorporating these changes and redrawing the design instantaneously.

In the area of *industrial design,* the airplane and automobile industries are leading the way in computer-aided design. The system used at General Motors allows the designer to build up from sketches on the display screen parts and details of an automobile such as fenders or roof sections. In the airplane industry, the design of cockpits, usually a costly and time-consuming problem, has been analyzed and solved by computers. Human engineering problems occurring in the operation of automobiles or airplanes can be examined and tested with a computer-drawn human figure which can be animated for each individual reach or operating position.

In *architecture,* computer programs have been written for structural analysis of steel and concrete, for heating or cooling systems and electrical distribution in buildings. But more important for the architect are the programs which assist him in the design and form-giving process. Computer programs which break down the design problem into smaller problems enable the designer to produce solutions to these subsystems before he attempts to solve the whole. The computer can be programmed to draw perspectives of buildings, interiors, etc., as seen from any viewpoint, thus helping the architect and planner to visualize buildings and even urban complexes, to modify the design and redraw it. This built-in trial situation is most important for the urban designer and architect because it will allow a new architecture, devoid of mistakes such as the opera house with bad acoustics or the highway interchange with no way to get on or off.

The computer will be very effective in the area of printing, publishing and *graphic design.* Print shops are rapidly acquiring computerized or tape-controlled typesetters and electronic color separators. Photo-typesetting is already partially replacing the traditional hot metal typesetting, and with the use of computer-generated typefaces, any typeface once stored in the computer can be reproduced with speeds up to 600 characters per second. The cathode ray display tube and the lightpen, a photo-sensitive stylus with which the designer, or editor in this case, makes his corrections, will revolutionize the newspaper and magazine field. The graphic designer, layout artist and typographer will be most directly influenced by such techniques as computerized typesetting, computer-generated images and new printing methods.

Computer languages have been written for the animation of pictures and diagrams, and *motion pictures* have been made utilizing the visualization possibilities of the computer used with a microfilm recorder to film the computer-generated images frame by frame. Examples produced at the University of California and at the Bell Telephone Laboratories demonstrate not only that this is possible but that it is economical as well. The added dimension of motion, important for any sequential representation of a design situation produced by the computer, has been put to use by the Boeing Co. in a film on aircraft carrier landings. This certainly opens up new vistas for the film maker, whether he is concerned with animated line drawings or complex tonal motion pictures — even color.

Computer-produced *abstract pictures* have already been exhibited in art galleries, but computer technology has only recently begun to play a more important role in art. In addition to producing linear compositions and op-art-like patterns, computer technology has been used in electronic sculptures whose moveable parts respond to each other and to the audience viewing it, and also in structures which, if activated by the museum goer, move and produce musical tones.

How will the computer affect the design profession and our environment? Very much like the children who are caught between the old math and the new, today's designers have to face the computer age, turn away from the security of the familiar and learn to adapt the new methods. Furthermore, in order to avoid the computer specialist solving the designer's problems, the designer will have to involve himself in this computer technology. Design educators, too, have to take a close look at the use of computers in design and re-examine their curriculum because clearly these are tools which can extend the abilities of the designer in almost every discipline.

Extensive changes in our environment will occur through computer technology. Within the next decade there will be a tremendous cut in costs for computers, with a great increase in computer application, making it possible to reduce the mass-produced uniformity started by the Industrial Revolution. In reality computers will not rob man of his individuality but rather enable technology to adapt to human diversity. The visual aspects of our environment will benefit from the use of the computer's ability to preview and "pre-test" the design itself and how it relates to its environment. Because all the computer visualizations and the tooling processes are backed up by mathematics, improvements in quality and precision, too, are likely.

The acceleration of production, information, learning and communication brought on by this new technology will not only change and upgrade our physical environment but also improve the quality of our life. The resulting changes in living patterns will again present questions and problems to be solved by the scientist, planner and designer. P.S.

GLOSSARY

The following is a list of computer terms, presenting part of the basic vocabulary of computer technology.

Accumulator: the part of the computer that stores results of arithmetic or logical operations.

Address: a number, name or label identifying a specific location within a computer's memory apparatus, or identifying a peripheral device.

Alphanumeric: pertaining to a set of characters that contains both letters and numerals.

Analog: denotes the use of physical variables, e.g., distance, rotation, or voltage, to represent and correspond with numerical variables that occur in computation; contrasts with Digital.

Analog computer: a computer that operates on analog data by performing physical processes on these data.

Binary: a numbering system using only the digits 1 and 0. A binary choice is one made between two alternatives.

Binary notation: the writing of numbers in the scale of two. Numbers zero to eleven are written as 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011. The position of the digits designate powers of two; thus, 1010 means 1×2^3 or 8; 0×2^2 or 4; 1×2^1 or 2; and 0×2^0 or 1. This equals one 8 plus no 4's, plus one 2, plus no 1's, or 10.

Bit: an abbreviation of Binary digIT (either 1 or 0).

Cathode ray tube: a vacuum tube in which cathode rays usually in the form of a slender beam are projected upon a fluorescent screen that serves as an anticathode where the rays produce a luminous spot. Sometimes called display tube.

Character: one of a set of elementary marks, such as numerals or alphabetic letters, or events, which may be combined to express information. A character includes all the marks, such as a group of holes punched in a tape, which are necessary to completely identify it.

Compiler: translates a problem from arithmetic statements as written by the programmer into machine language instructions as understood by the computer.

Computer: a device capable of solving problems by accepting data, performing prescribed operations on the data, and supplying the results of these operations.

Console: location of computer controls, as well as various lights, cathode ray tube display.

Coordinates: the positions or relationships of points or planes.

Data: any facts or information, particularly as they are taken in, acted upon, or emitted by a machine used for handling information.

Decimal digit: one of the numbers 0 through 9 when used in the scale of ten.

Digit: one of the symbols 0 through 9 when used for numbering in the scale of 10, regardless of position or the type of Code in which they appear.

Digital computer: a computer which produces results from numeric information only, and performs operations by means of counting, rather than measuring as in analog computers.

Disc: a set of magnetic plates on which information is stored on both sides.

Drum: a magnetic cylinder on which information is stored.

Hardware: the physical assembly of the computer and its accessories, as distinguished from the programs known as "software."

Input/Output (or I/O): devices used to connect the computer with the operator; e.g. printers, tapes, card readers, etc.

Iteration: repetition of a small series of simple steps to perform difficult calculations.

Keyboard: part of a device that punches holes in a card or tape to represent data, or a device that communicates directly to a computer.

Library: set of useful routines stored within the computer, available to all users.

Lightpen: a hand-held pen-like device containing a photocell or photomultiplier, used for guiding the generation of lines on the display.

Machine language: instructions written as binary codes.

Memory: a term referring to the equipment and media used for storing information (data and instructions) in machining-language electrical or magnetic form.

Numerical control system: a system in which actions are controlled by the direct insertion of numerical data. A numerical control (N/C) machine must automatically interpret at least some portion of the data.

Object program: assembled or compiled program in machine language.

Offline operation: operation of peripheral equipment such as card readers and magnetic tapes independent of the central processor of a computer system.

On-line operation: operation of input and output devices interacting directly with the central computer.

Optical scanner: a device that optically scans printed or written data and generates their electrical representation for input to the computer.

Plotter: automatic drawing equipment controlled by a tape or directly by a computer.

Printer: an output mechanism which prints or typewrites characters.

Processor: that portion of a computer which controls the operation input and output devices and operates on the received, stored, and transmitted data. Its circuitry includes the functions of memory, logic, arithmetic, and control.

Program: a set of instructions for the computer that defines a desired sequence of conditions for a process or function, and the operations required between these conditions.

Punched card: a card of constant size and shape, suitable for punching in a meaningful pattern and for mechanical handling. The punched holes are usually sensed electrically or mechanically.

Punched tape: paper tape into which a pattern of holes is so punched as to convey information.

Random access: indicates equality of access time to all memory locations, without dependence on the location of the previous memory reference.

Real time: computer takes data, makes decision, and responds with solution within same time span as real life phenomena.

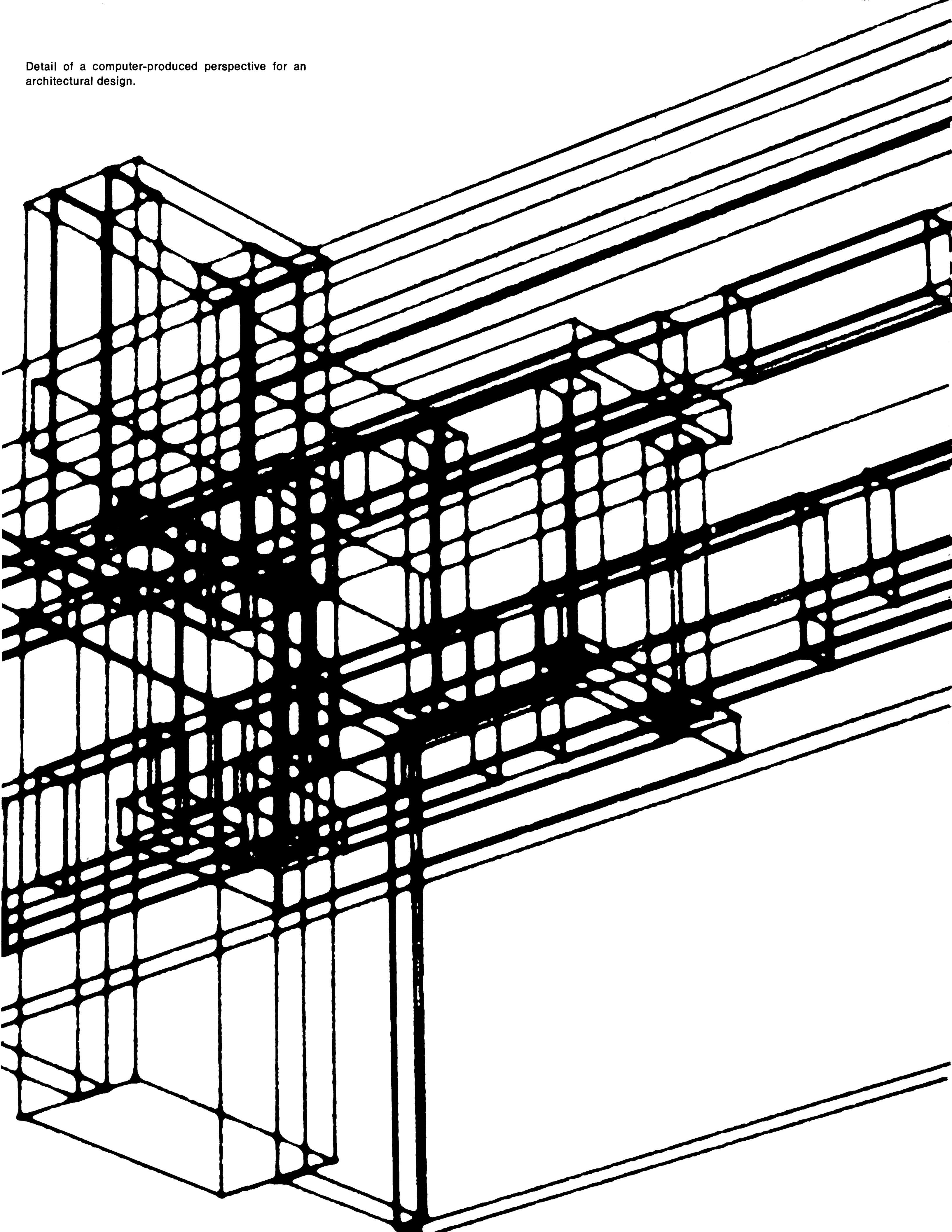
Register: part of computer storage device where data may be operated on.

Routine: a set of instructions arranged in the correct sequence to direct the control computer to perform one or a series of operations. A portion of the total program.

Software: program; the means of communication with the machine as distinguished from the hardware.

Time-sharing: process in which computer switches rapidly from one problem to another, giving to each of a number of human users the illusion of working upon his problem all of the time.

Detail of a computer-produced perspective for an architectural design.



Stephen A. Coons is Associate Professor of Mechanical Engineering at the Massachusetts Institute of Technology, where he is in charge of the computer-aided design group. He was previously design engineer for the Chance-Vought Aircraft Division of the United Aircraft Corporation where he developed mathematical methods for describing the shape of aircraft fuselages by computer. Professor Coons is author of numerous papers and articles on design, the geometry and description of shapes, and computer graphics.

In the following article, Professor Coons analyzes the human and mechanical aspects of the creative process and proposes to let the machine, in this case the computer, take over where the task becomes repetitious and non-creative. In the description of Sketchpad, an early man-machine graphical communication system, Professor Coons demonstrates how drawing is used as a means of communication with a computer for the purpose of design and engineering. This system contains input, output and computation programs that enable it to interpret images directly drawn on a computer display. For instance, it is possible to draw a basic bridge shape onto the display tube, a television-like display, and then experiment with various loading conditions and different supports. The computer produces a visual display showing the changes resulting from the application of different loads and the effects of minor modifications of the bridge design.

In a further development of this system it is possible not only to work with two-dimensional but also three-dimensional images, which, of course, opens this system to the industrial designer, architect and graphic designer.

COMPUTER-AIDED DESIGN

by Steven A. Coons

HUMAN AND MECHANICAL ASPECTS OF THE CREATIVE PROCESS

While designers in engineering, perhaps, are less interested in aesthetics than designers in other fields, all creative designers are involved in a similar process. This design process unfolds something like this: at the beginning, in the design of a device or system (be it a motion picture projector, an airplane, an automobile, or a battleship), the designer does not have a very clear notion of what he wants to do. He has only a vague concept, or none at all, of how he will go about accomplishing his task. In this sense, the design process is a learning process during which the designer must learn what the problem is and how to solve it. Within this process of learning there are certain exciting aspects of discovery. But these are interspersed with long tedious periods of rote behavior — sheer unadulterated dull work — noncreative but necessary. It is appropriate to have computers to do this noncreative work so as to leave the designer free for the activities human beings are good at: innovation — the association of hitherto unrelated ideas. The typically human aspect of the design process is invention: the grasping of schemes that are at the beginning vague, tenuous, dream-like, and solidifying them into something tangible that can be looked at, explored qualitatively, and evaluated quantitatively. To the same schemes, one can apply analytical procedures and then, on the basis of these procedures, make more precise judgments. While all activities during the design process up to the application of analytical procedures are humanoid, analytical procedures are essentially not.

As another example: mathematics, however much a man may enjoy it, is to a large extent a mechanical and not, strictly speaking, a human process. Human beings cannot follow through step by step every detail of a mathematical proof. They cannot multiply two ten digit numbers together without, in each case, a piece of paper and a pencil in order to temporarily store some of the details involved and think up some more details to write down. On the other hand, the strategy by which we prove a mathematical problem can be dealt with in a man's head without pencil, paper or other artificial mechanical means of storage. There is a dichotomy between what people can do with ease and what is mechanical and therefore should be done by machine. The details of a mathematical exercise, whether it be a proof or an execution of a computation, are usually too cumbersome and complex for human creatures to remember and should be done by machine, while the construction, or the structuring of the strategy that yields the mathematical exercise, is a process that people can and should do. Mathematics, as design, is an example of a creative process.

While engineering designers have often less to do with aesthetics and more with mathematics, even mathematicians have some sense of the aesthetic in their proofs. Actually, designers in engineering, architectural designers, industrial and graphic designers, whether they pay a great deal of attention to aesthetics or not — even mathematicians and physicists — all of these people are involved in a similar process in that they, at the beginning, structure ideas, form concepts, produce associations, examine tentative trials qualitatively, behave as generalists of ideas and then subsequently test these ideas by various techniques — mathematical, computational, mechanical.

REQUIREMENTS FOR MAN-MACHINE INTERACTION

In order to make a computer an assistant in the design process and in order to make it do the part of the design process that is noncreative work, several requirements must be met. The problem is to make it possible for a designer and a machine to work on problems together — the designer doing what the machine can't do and the machine doing what, in a sense, the designer can't do (like evaluating the product of two ten digit decimal numbers). One important requirement is that the designer can talk to the machine in a natural way, using natural forms of communication. Natural forms of communication with a computer, or better still, natural ways of communicating ideas and information include, of course, the graphical form.

Another important requirement is that the designer can interact with the computer as though the computer was paying strict attention only to him. But this must be economically feasible because computers and their services are very expensive. As a rough but conservative estimate, it costs $10 a minute to use the computer. In the creative process, the designer may sit vis-a-vis a computer and while away time thinking. It would be a waste of money to have the computer wait patiently for the designer to think up a new idea. Consequently, as of recent years, a new concept of time-sharing a computer has been developed. This means that the designer can sit with the computer, "holding its hand," so to speak, and talk to it, while the computer will seem to pay attention only to him. But, in fact, the computer pays attention to the designer for a very short period of time and then turns away from him and looks at some other user, pays attention to him, and then turns away from him and pays attention to someone else, and so on. The computer may pay attention to as many as thirty, forty or one hundred different people who are talking to it, communicating with it, having it work on their problems. It executes or takes appropriate action in response to their commands, wishes, and their communications so rapidly that each individual user believes that he has the undivided attention of the machine. In this way the computer is kept constantly busy and all the users of this system are busy and happy and don't have to feel that they are wasting money if they do wish to sit and think.

To sum up the idea of man-machine interaction: the designer sits at some kind of a terminal device — a console — connected to a computer system. There are many such consoles with many users seated at them working on design or other problems and sharing computer time. The interaction takes place through all forms of communication such as graphics, mathematical symbolic statements or ordinary English.

Dr. Ivan E. Sutherland working on a Sketchpad display tube mounted on the console of M.I.T.'s Lincoln Laboratory TX-2 computer. Push buttons used to control specific drawing functions are on the box in front of him. Size and position of the part of the picture seen on the display tube is regulated by means of the four black knobs below the screen.

THE ORIGINAL SKETCHPAD SYSTEM

An important first step toward *graphical* communication between the designer and the computer was Ivan Sutherland's "Sketchpad" program written for the TX-2 computer of the Massachusetts Institute of Technology, Lincoln Laboratory. This program was completed in 1962. In computer technology, something that is four years old becomes very quickly worthless, including computers themselves. In this sense a four-year-old computer item is certainly an "antique". From this point of view the original Sketchpad system appears almost as a remnant of the past.

But, nonetheless, this system could do some things which were very sophisticated. It had a console so complex and confusing that it could startle and frighten any operator: toggle switches activating programs for "erase," "start drawing," etc.; dials; blinking lights; a light pen; a 7 x 7" cathode ray tube, etc. — but this is not too important. An early radio, too, had numerous knobs and was completely open so that the tubes were exposed. In order to receive a station, all the knobs had to be adjusted in a kind of trial and error fashion and there was no clear, descriptive notion about what to do in order to tune in a station. All the dials were, in a sense, cross-coupled. The original Sketchpad system bears the same relationship to a modern console as an old fashioned radio with all its knobs to a modern television set.

Sketchpad was purely graphical and purely geometrical. One could draw with the light pen on the screen — straight lines, circles and other surfaces. It could not solve any problems that had to do with abstraction, other than the abstractions of geometry itself. The operator could impose constraining relationships such as: "Make these two lines parallel" to the computer. That, of course, is a geometrical abstraction and the computer program could follow such an instruction. But the designer could not say to the early Sketchpad: "This line represents a piece of structure with a certain thickness and with certain cross-sectional characteristics, made of a certain material and obeys certain physical laws."

Sketchpad had toggle switches that commanded the computer to satisfy such constraining relationships as the operator imposed. There were four knobs beneath the screen of a cathode ray tube which were used for four kinds of motion applied to the drawing on the screen: a rotation, a horizontal translation, a vertical translation and a change in magnitude. The precision of the graphic information presented on the screen is one part in ten million which means one thousandth of an inch in eight hundred feet! This made it possible to enlarge, look at a small region, change it and then push it all away from the viewer (by reducing its size) so that the entire structure could be visualized.

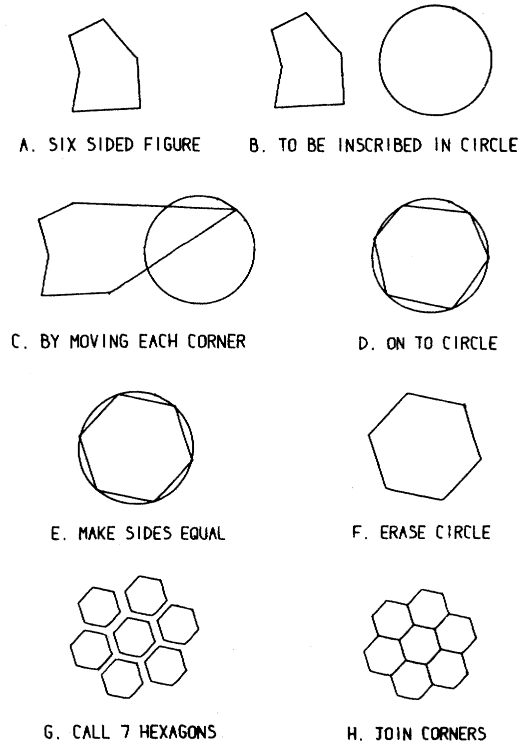
Sketchpad could attach one line to the end of another separated line by means of toggle switch instruction, even if the light pen would not attach the lines precisely on the screen. The computer interpreted the instruction in such a way that the two lines were attached with mathematical precision at the end point, so that they were truly concurrent.

The foregoing example shows that the computer can interpret meaning. The interaction between operator and computer goes somewhat like this: the computer "says" to itself: "I think what the boss means is that these two lines should be concurrent." Therefore, subsequently, if "the boss" pulls the figure apart, the computer will put it back together again. Now the computer has many ways of reassembling the figure. It does not know exactly in what way "the boss" wants the lines attached, it only knows that it has been told to attach them. So it will choose, automatically, one specific way of attachment and if that turns out to be appropriate — fine. If the operator does not like the decision the computer makes, he can talk with the computer and say: "No, I didn't mean that, I meant something else." In other words, it is not necessary nor even desirable in a man-machine interactive relationship that the computer be taught to consider all contingencies and pick the best one. It is only necessary that the computer make some response. Then the operator can interact with the computer and modify the response in a direction that is appropriate to his purpose. This is very much like the psychological process called "reinforcement." In this case the behavior of a machine instead of an organism is reinforced.

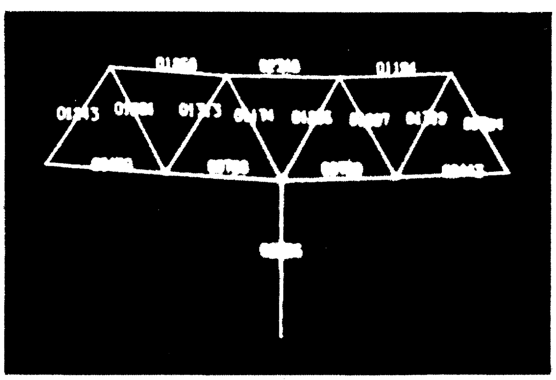
To the extent that a design problem was of geometric nature and did not involve problems requiring expressions in non-geometrical terms, Sketchpad could solve design problems. The operator could, for example, draw a cross-sectional picture of a little angle bracket on the cathode ray tube with the light pen and then "drill a hole in it" by adding the appropriate lines to indicate the hole. Then he could draw separately a rivet, play with the shape of the rivet until it suited him, call for the bracket and for the rivet at the same time and then put both together.

ARCHETYPES AND ICONS

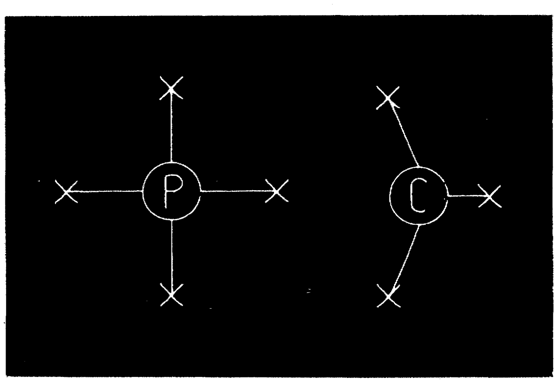
Although with the original Sketchpad only straight lines and circles could be drawn, it



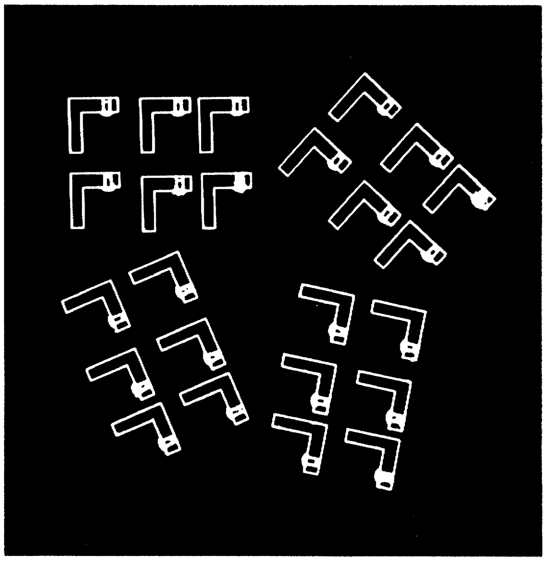
A pattern of hexagons produced with Sketchpad. The designer points the lightpen at the display and presses a button called "draw" and the computer will construct a straight line stretching like a rubber band from the initial to the final location of the lightpen. Repeated pressing of the button will produce additional lines forming, for example, an irregular six-sided figure. In order to make this figure a regular hexagon, the designer can inscribe it in a circle. The circle is drawn by pointing the lightpen at the screen and pressing the button "circle center." This leaves a center point on the screen. Choosing a radius, pressing the button "draw" again and moving the pen in an approximate circle, the designer can cause the computer to construct a perfect circle with the chosen radius. Using similar operations, the corners of the six-sided figure are moved one by one into the circle, its sides are made equal, the circle is erased, the number of hexagons is automatically increased and finally joined at the corners.



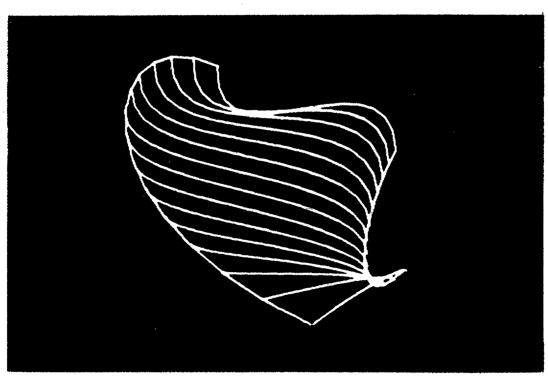
Display from Sketchpad System showing stresses resulting from application of load via lightpen to a geometrical representation of a bridge truss.



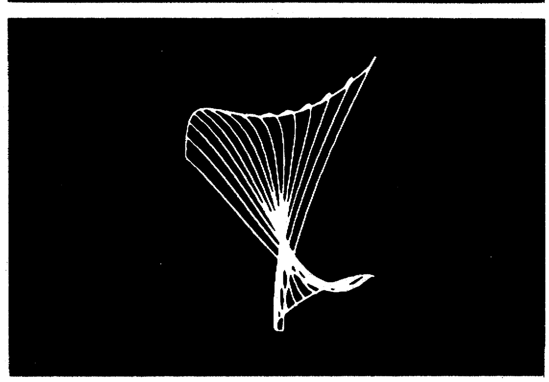
An "icon" or display of a constraint such as "parallelism" (P) or "circle" (C), can be called forth on the screen to tell the operator that the computer has applied a constraint.



Compounding of arrays of similar objects. Once the fragment is designed it becomes an "archetype" which, if stored in the computer, can be replicated any time. The display shows four arrays each consisting of six identical elements to make a compound array of 24 fragments.



Two views of a free-form surface as generated by the computer. The design information consists of the four boundary curves; the computer supplies the smooth internal contours in about one tenth of a second. The surface can be modified in any way the designer sees fit.



A free-form surface rotated in space to reveal its shape more clearly to the designer.

embodied several very important ideas. One was the principle of "archetypes." If one were to draw a rectangle, using straight lines, then it would be never again necessary to construct a rectangle, because the simple rectangle, once drawn, becomes the generic archetype of all rectangle-ness for future use. Likewise, if one designed a "thing" (such as a rivet), and that "thing" was important enough to be used again and again, it would become in some sense an archetype and one would never have to design it again. It then would exist in the computer and could be replicated at any time.

Another important idea used in the original Sketchpad system was the notion of constrained relationships. In any random sample of two lines drawn haphazardly (as if one were to throw thin sticks on a surface), most pairs of lines will be intersecting. If the operator wanted to make them both parallel and equal in length, he could communicate his wishes to the computer by calling for two atomic constraining relationships. The first was that the lines be parallel, and the second, that they be made equal in length. When the operator did call for these constraining relationships he certainly could remember whether he had or had not called for these two constraints. But not all situations might be that simple — indeed the operator may wish to impose many other atomic constraining relationships. Therefore, since there is a potentially very complex situation, the computer should be able to exhibit an "icon," a graphic symbol, which stands for the atomic constraining relationship. The icon should be a sign which tells the operator that the computer has thought of and applied a constraint. At the same time it should also communicate to the computer that the constraint exists. The icon should have transparency — allowing the computer to look out at the operator and see his wishes and allowing the operator to look into the computer and see that the computer has paid attention.

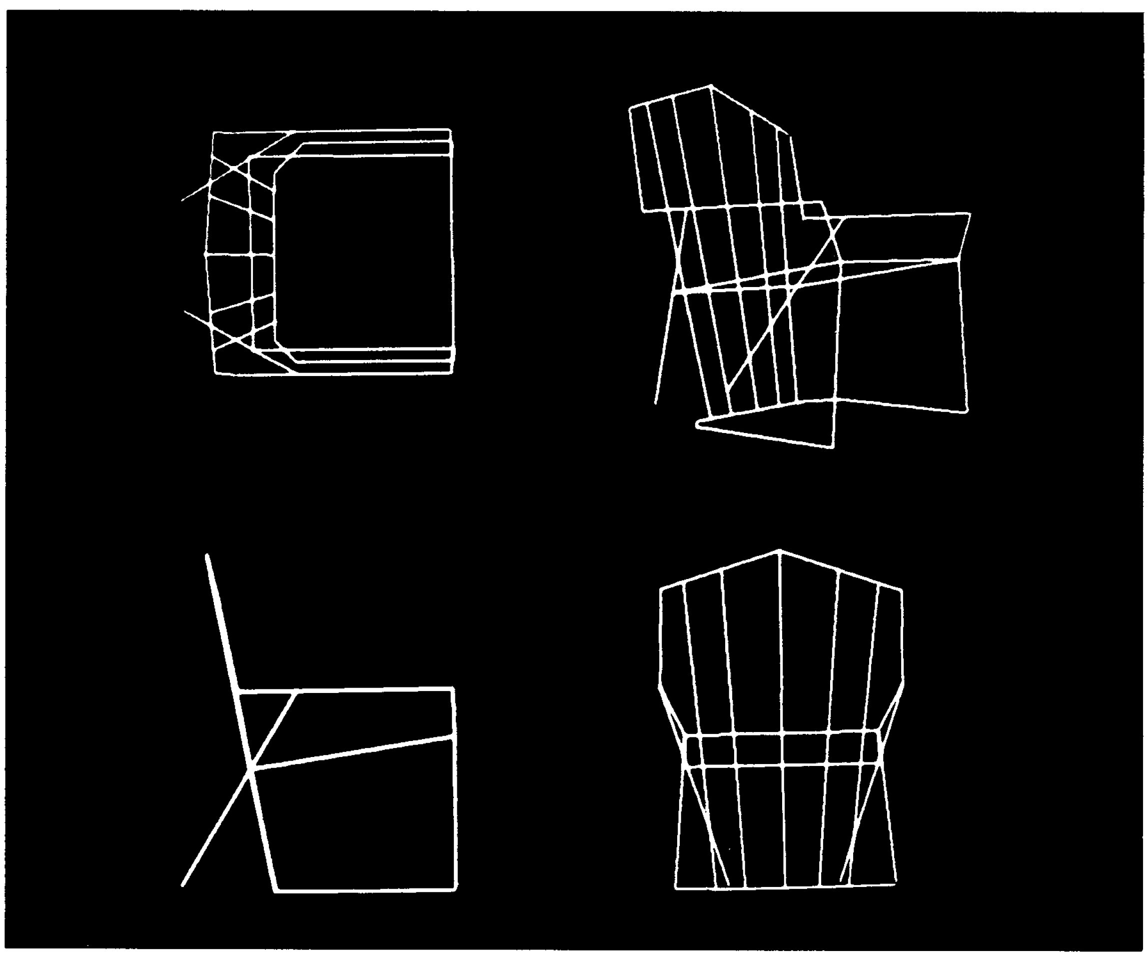
The original Sketchpad had such an icon for making lines parallel and another icon for making lines equal in length. There were other icons for making lines horizontal or vertical when they were drawn nearly horizontally and vertically by the light pen. There were nineteen such constraining relationships capable of being called for by the operator. In today's system, there are a great many other constraints and relationships that can be invoked, but these nineteen served in the past when Sketchpad was written. Although the notion of making a graphic icon to represent constraints is not done today in the same way Ivan Sutherland thought about it, icons are still used, and constraints are applied to the objects or the elements of pictures produced in such systems. In this sense it is very appropriate to credit Mr. Sutherland with the scheme.

Timothy Johnson working on a Sketchpad 3 display.

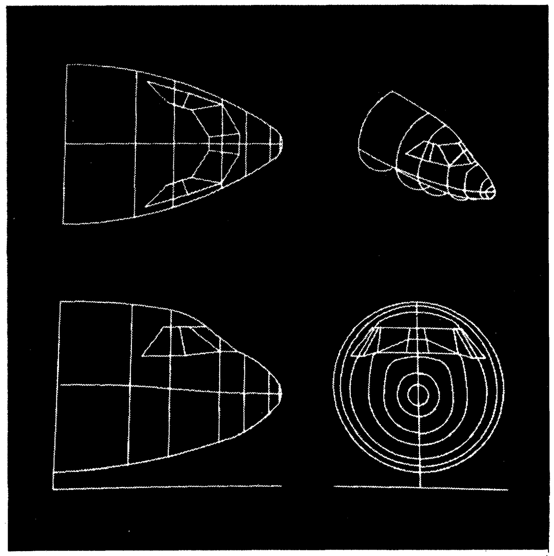
SKETCHPAD 3

The classic Sketchpad system was followed by "Sketchpad 3." This name indicates an extension of the two-dimensional Sketchpad into three dimensions. Sketchpad 3 allows direct communication with the computer, which results in three-dimensional images. For instance, the operator can see a three-dimensional surface, and furthermore, have it rotate in space. In one sense, Sketchpad 3 is less sophisticated than Sutherland's original version which was a new idea. But in another sense it is more sophisticated. After all, it is easy to draw two-dimensionally, but it is not so easy to teach the computer that there is a three-dimensional space and that the cathode ray tube is not the world or the universe, but that there is something beyond it.

Sketchpad 3 was developed by Timothy Johnson, who, at M.I.T., used the basics of Sutherland's Sketchpad and built on them. In his system the face of the cathode ray tube is divided into four quadrants: one quadrant is for the plan, one for the front view, one for the side view and one for the perspective view of any object one might want to draw. If the plan and the two elevations are drawn, the perspective view of the object will "automatically" appear in its quadrant. Likewise, the conse-



Sketchpad 3 display showing three orthogonal projections and a perspective view of a wire-frame chair.



Display from the System Science Section at the Lockheed-Georgia Company, Marietta, Georgia, showing top, side, front and perspective views of the forward fuselage of a large transport aircraft.

quences of any change the designer makes with his light pen in any one of the four quadrants are "automatically" shown in the other three quadrants.

Sketchpad 3 allows for rather complex visualizations. For example, a mosaic-like pattern wrapped around the surface of a sphere would be very tedious to construct with pencil and paper. To do it with Sketchpad 3, the designer would use his light pen and switches as follows: he would start with a vector and a fixed point and establish a drawing plane on the right hand end of that vector. The vector becomes a way for visualizing an invisible sphere. Then he would draw diamond shapes. These would be attached (by pulling them into place with the light pen) to that invisible sphere. If the designer were to decide on a smaller diamond, he could erase the original and then draw one in the desired size. Next he would rotate the front view about an axis to put the invisible sphere into a new position. Now with the sphere in a new position he would draw another diamond. Then the sphere is rotated again. At this point he could stop drawing diamonds because the computer now knows how to go about drawing diamonds and the designer has an inexhaustible supply of prototype diamonds and a way of attaching them. Two rotations have taken place and the sphere can now be put in any general position in space.

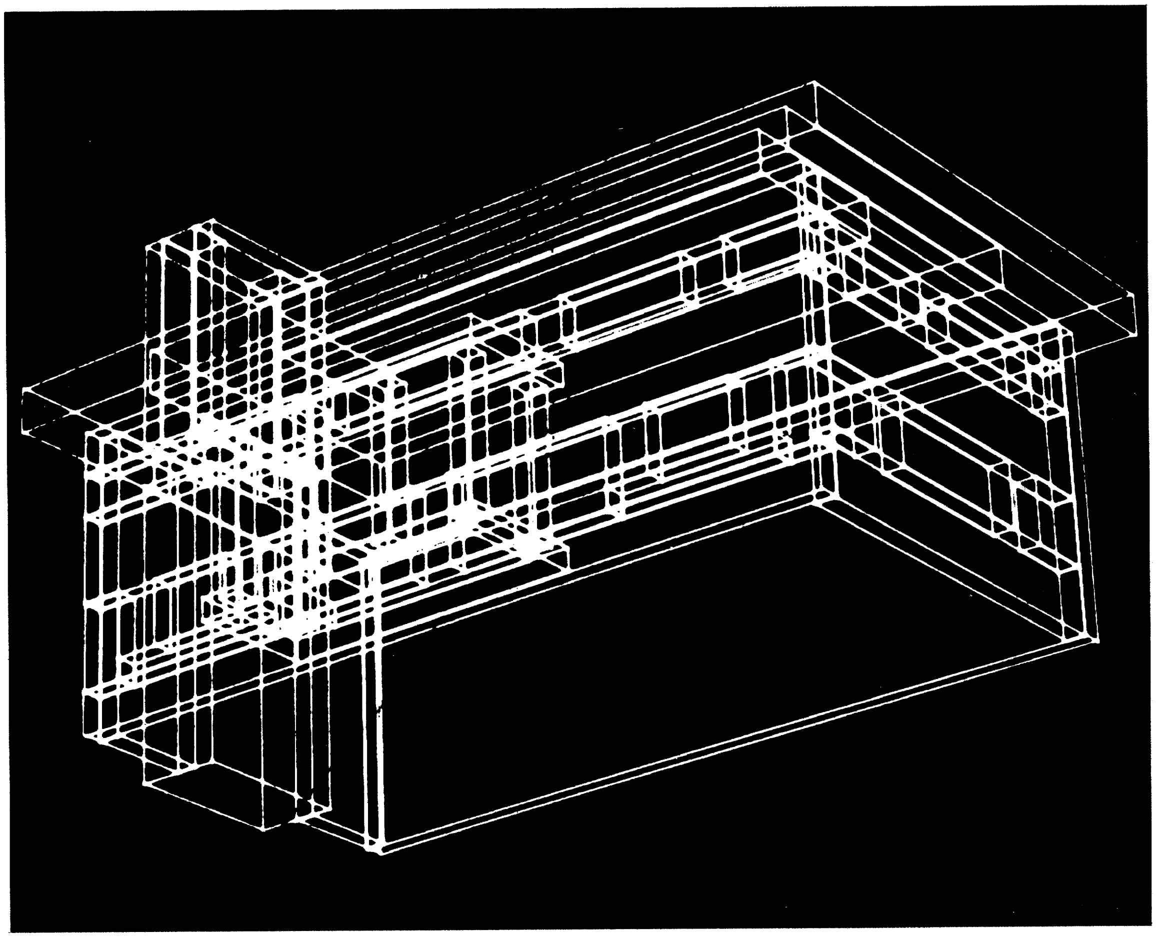
The computer does not know that a sphere happens to be a relatively simple surface. It turns out that the mathematics or the representational structure inside the computer does not distinguish between the complexity of that particular surface and that of a battleship, an airplane, or a tobacco pipe. Today one can draw virtually any kind of a shape: the hull of a ship or even freehand sketched curves. It will not be long before the designer can literally do sculpture with a computer. He will be able to draw three-dimensional configurations as complex as the human figure. The computer will then have an internal structuring of information so precise and so detailed that it will be able to probe any independent point on the surface of the three-dimensional human figure that exists only in the computer's data structure, and to determine its surface with great precision.

NON-GEOMETRIC ABSTRACTIONS

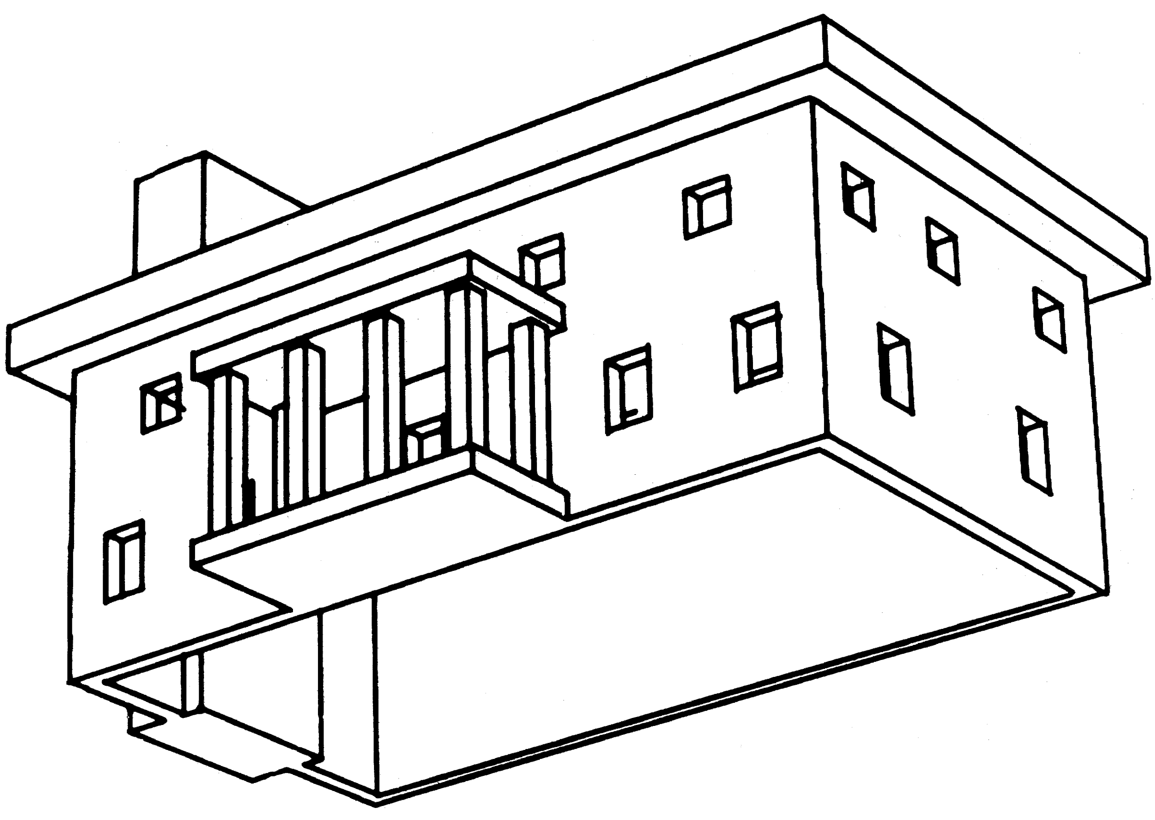
With the original Sketchpad system it was not possible to solve any non-geometrical problems. Today's complex systems, however, are able to solve problems containing other than geometrical abstractions. The designer can draw a diagram which does not represent a "thing" but an "idea." This idea might be that there is a number and another number and a process adding these two numbers together. This may be drawn with the light pen on the cathode ray tube in the form of a diagram. The computer's output, the result of that drawing procedure, could be stored or some other mathematical operation could be applied to it. The diagram would show procedures, inputs and outputs of procedures, and tie all this together. This amounts to putting numbers in one end of this diagram, causing the computer to interpret the meaning of the diagram, take the numbers, perform the indicated operations on the numbers and yield the result. Such a diagram would not be a graphic representation of an object, but of a totally abstract system.

A designer can also draw a diagram that may be an electric circuit of any degree of complexity. He then can describe to the computer what each of the elements of this electric circuit means. Having done that, he can find out what the resulting currents and voltages are, across and through various pieces of this diagram. He also can draw a picture of a thing consisting of pure geometry, and then assign to the geometry other kinds of abstractions which are non-geometrical so that geometry and abstractions are mixed together in any desired way but presented visually on the cathode ray tube.

The original Sketchpad system thus introduced the basic tools for computer-aided design and established some of the principles of a man-machine interaction. Sketchpad 3, with its development of a three-dimensional presentation of images, surfaces and shapes, increased the possibility of applying this potential to many more areas of design and engineering. Today, with much more complex and efficient computer systems available, the designer is not only involved in the innovative function of design and the mathematical process of analysis, but by transmission of the final design solution directly to the tooling process he can also affect the final manufacturing of the product in question.

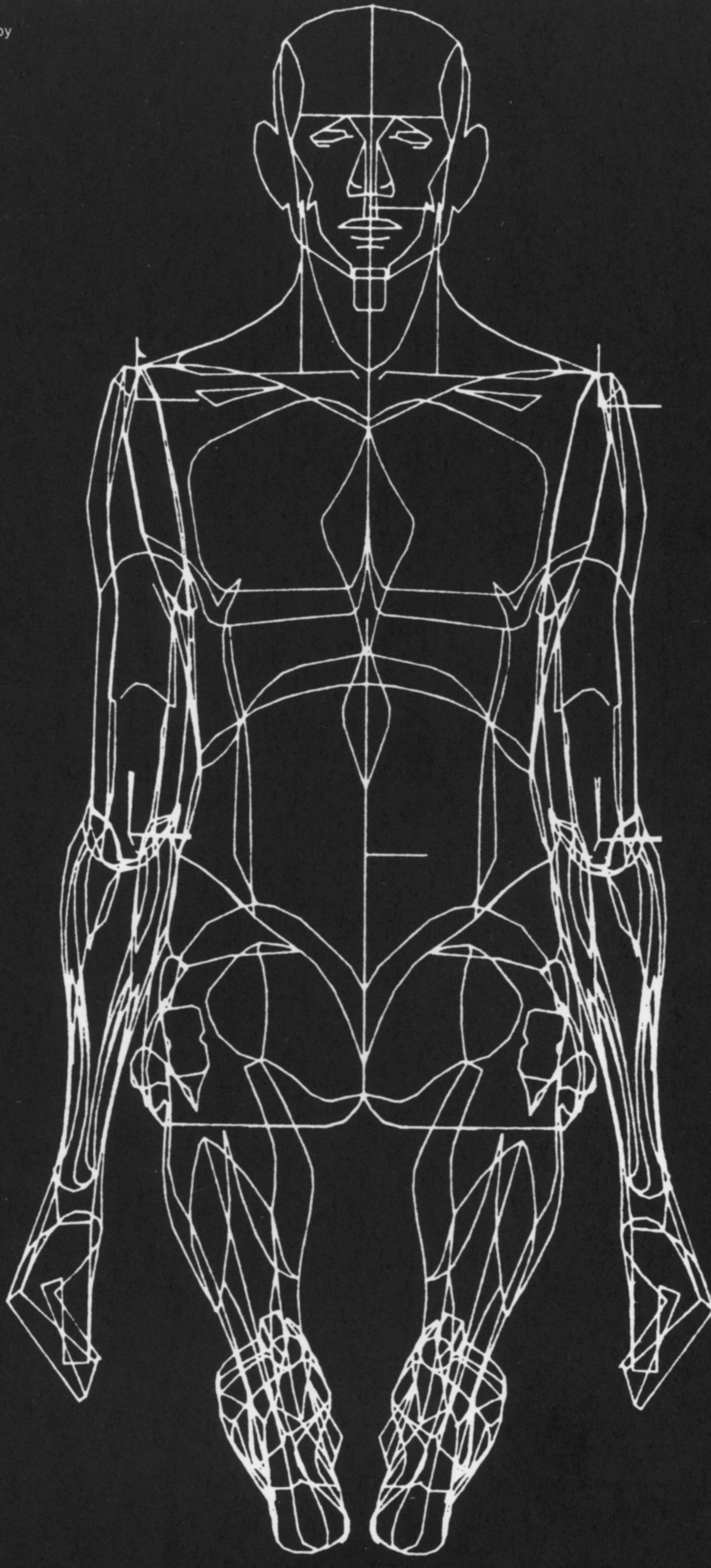


A perspective view of an architectural design. A program can be prepared that will cause the computer to display every part of an object. In the illustration below, lines that would not normally be visible have been removed.



It is difficult to write a program that will cause the computer to remove the hidden lines from the display. Even with an appropriate program it takes the computer a relatively long time to carry out the instructions. The program for this display was written by Lawrence G. Roberts of the Lincoln Laboratory at M.I.T.

Animated human figure drawn by the computer.



William Fetter is supervisor of Computer Graphics at the Boeing Company in Seattle, Washington, where Computer Graphics was developed and made available as a tool for the visualization of the many problems related to airplane production. Fetter's department, made up of graphic designers, illustrators and writers, prepares and produces publications, documents, slides and motion picture animation for the development and promotion of new products.

Computer Graphics is the technique which uses the computer to produce still or moving images, whether on paper, tapes or film. In 1960 Fetter outlined a new concept of perspective which abandoned the accepted academic drawing methods. Fetter and Walter Bernhardt converted this concept into mathematics. A programmer then translated the concept into computer language. Data from an aircraft drawing was supplied to the computer connected to a plotter (automatic drawing equipment) which delineated a perspective drawing.

This first perspective drawing produced by a computer already indicated the possibility of using computers in many areas. In fact, during the past few years a whole new area of graphics has been created with the application of Computer Graphics to many visualization problems. The examples shown in the following article by William Fetter demonstrate how Computer Graphics can be applied to the visualization of acoustical graphs, the evaluation of preliminary cockpit designs, the location of radar stations, the design of a cockpit display system and the simulation of an aircraft carrier landing.

Recent developments of more sophisticated equipment for computer-generated images increase the area of Computer Graphics while eliminating most of the tedious work that is usually required. It is now up to the designer to use Computer Graphics and thereby extend his ability to solve complex visual problems.

COMPUTER GRAPHICS

by William A. Fetter

Computer Graphics represents a new stage in the art of visual communication. Several thousand years ago the Babylonians recorded their accomplishments with stylii on tablets in characters not unlike those punched in IBM cards. Early printing was a technique for reducing the laborious and expensive process of writing on paper. Present-day typesetting equipment, related photomechanical processes, and the modern printing presses have further simplified the translation of thought into visual form.

Through Computer Graphics visual communication has developed into a technology with computers converting virtually unlimited engineering or scientific data into visual images. Just as in the past, however, this latest development has brought new standards, changes in the accompanying human crafts, and even greater demands upon human judgment and management in operating the machines, facilities and tools involved.

Perhaps the best way to define Computer Graphics is to find out what it is not. It is not a machine. It is not a computer, nor a group of computer programs. It is not the know-how of a graphic designer, a programmer, a writer, a motion picture specialist, or a reproduction specialist.

Computer Graphics is all these — a consciously managed and documented technology directed toward communicating information accurately and descriptively.

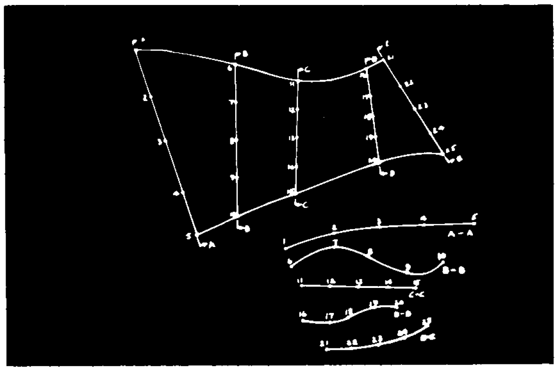
The term "Computer Graphics" was coined at the Boeing Company in 1960. It was obvious that comprehension through the human eye was the primary goal in employing computer technology to visualize complicated relationships and images. Special graphic skills were required to meet these goals and only the utilization of the maze of electronic circuitry in contemporary computers, along with the necessary computing skills, could help to produce graphic images economically and rapidly.

In Computer Graphics, three aspects have to be understood: first, the communicator who has an idea or message to communicate; second, the communication specialist who decides whether the problem should be solved graphically, verbally or as a combination of both; third, the computer specialist who selects the computer equipment and interprets the problem so that the computer can act upon it. The present make-up of the Computer Graphics organization at Boeing recognizes these aspects and thus provides a combination of creative communication and engineering accuracy for managers, engineers, and others in any Boeing division.

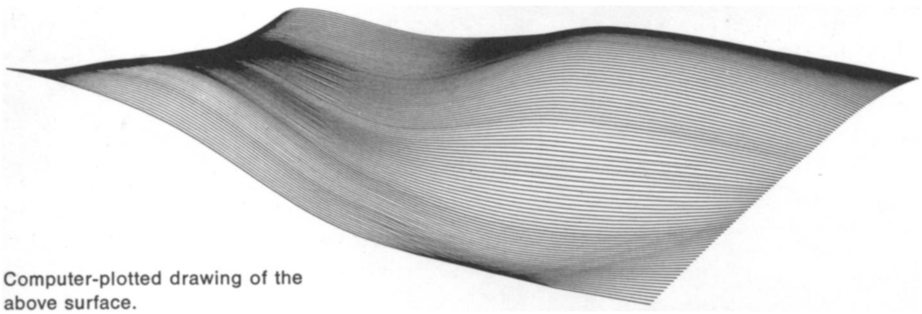
COMPUTER GRAPHICS IN RESEARCH

One important research area is found in the relation of Computer Graphics to Numerical Control of machine tools. Numerical Control is a system that regulates the action of one or more machines by the automatic interpretation of instructions which are made up in numerical form.

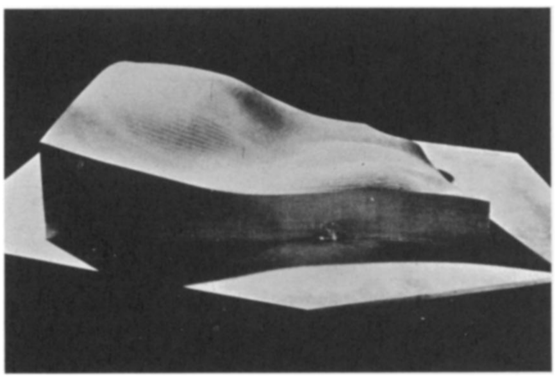
For instance, a particular surface is defined by selecting 25 points. The computer then provides a complete definition of the surface which then can be produced by employing Computer Graphics capabilities. To check its correctness or examine its acceptability, it can be presented as a true perspective or any other projection. Working from the same data, with some modifications for machine tool characteristics, a punched tape can be prepared which controls a machine that cuts this surface from metal. It is obvious that here is a great potential for producing many items necessary to aircraft manufacture. Other research activities in Computer Graphics result in films showing visually structural dynamics, that is, changes due to maximum stress, etc. With a plotter (automatic drawing equipment controlled by a computer)



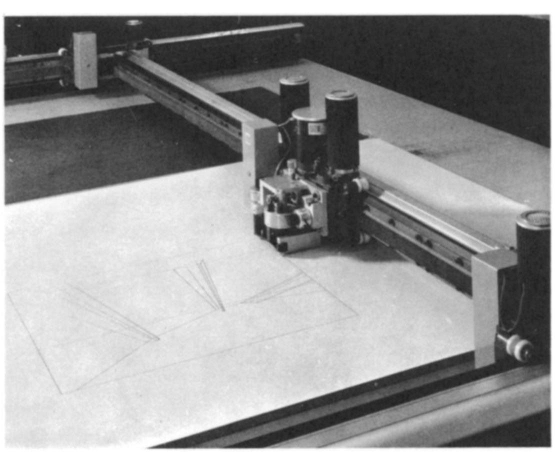
Surface defined by 25 points across different sections.



Computer-plotted drawing of the above surface.



Same surface cut into metal by numerically-controlled machine tools.



Automatic drawing produced by a plotter, which is either operated by a magnetic tape or directly controlled by the computer.

it is possible to produce pictures of an airplane for a film in which the wings assume different positions because of particular flutter modes. Instead of examining long listings of computed numbers, through the film the structural engineer can study visually and in motion the forecasted performance of a new design.

The results of this and other research are documented in procedures that match the work flow of applications — from authorization, communication design, data transcription, computer programming, automatic drawings and final rendering to reproduction. Long lists of skills, programs, equipment and systems are involved. Computer Graphics technology depends on identifying these elements, developing the means to manage them and, perhaps as important, knowing which pathway to choose through this maze.

The following is an outline of the procedure and indicates how Computer Graphics was applied at Boeing:

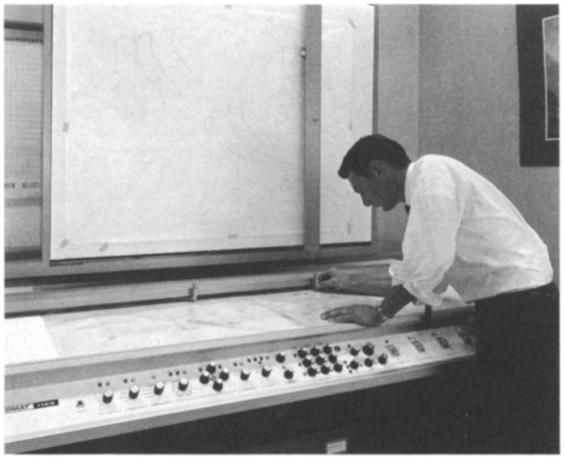
Authorization includes examining the application to determine if Computer Graphics is, in fact, the best method for solving the problem presented. If so, authorization papers are prepared to assure an agreement of the aims of the application.

Communication design is the key to the remainder of the Computer Graphics process. For instance, in the case of a motion picture, a storyboard is prepared by the designer. All subsequent activity, including computing, will be based on the communication sequence which the designer prepares with full knowledge of the capabilities, costs, and time available to him.

Data transcription, the process of transferring measurements into data acceptable to the computer, is accomplished by marking a transparent overlay on an orthographic view of an airplane, and selecting the numbers and the type of coordinate points indicated by the designer. This transparent sheet is given to a Telereader operator to complete the data transcription step. The operator places the sheet on a back-lighted screen, aligns cross hairs on the screen by two control knobs, and then produces a punch card by pressing a foot button for each point.

Computing and programming follows when the airplane is defined by a complete deck of cards. The cards are combined with Computer Graphics programming cards in the computer which then produces a magnetic tape.

Automatic drawing is the next step. The magnetic tape operates automatic drawing equipment such as the Orthomat, a plotter which produces drawings automatically. A computer/ recorder, which is usually used for motion



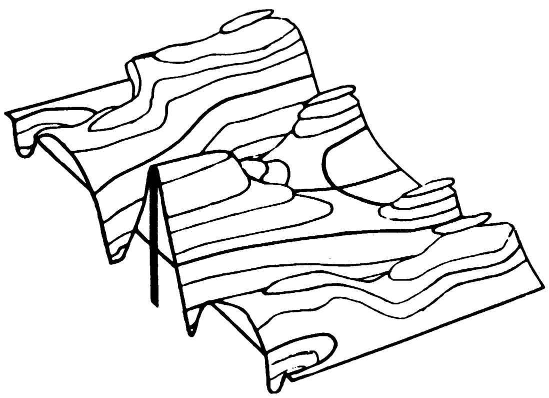
The Illustromat 1100, a computer-directed drawing instrument capable of automatically drawing perspective views.

pictures and documents, produces, instead of drawings on paper, a 4 x 4 inch image on a cathode ray tube, a television-like screen, which is automatically photographed on 16mm or 35mm film within the equipment. Many options are possible within these systems. For instance, the Illustromat 1100, a computer directed drawing instrument, combines the steps of data transcription (on a horizontal surface), computing programming (at a front console) and automatic drawing (on a vertical drawing board), and simultaneously draws any type of projection as the operator moves the stylii along coordinated lines on the horizontal surface.

Final rendering is the next step which may include illustrating the automatic drawings and the addition of lettering and animation in the case of motion pictures. Final rendering produces some type of master (either plates, negatives, or a motion picture master) from which the final step, reproduction in quantity, can be accomplished.

APPLICATIONS OF COMPUTER GRAPHICS

ACOUSTICAL ENGINEERING



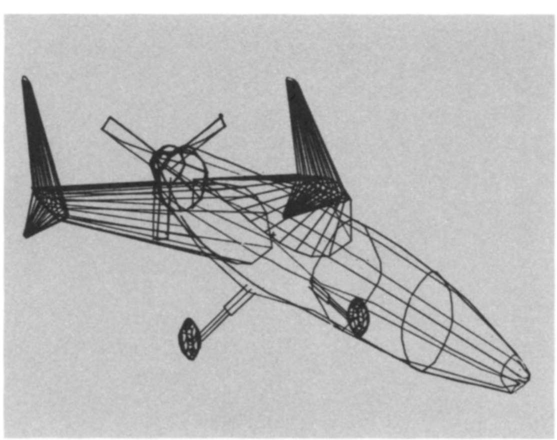
Three-dimensional acoustical engineering graph shown in perspective for better understanding.

The results of an advanced aerodynamic-acoustics study can be presented as a three-dimensional acoustical engineering graph. The graph can be shown in isometric projection and also in rotated perspective views to allow the engineer a better understanding of its topology.

ENGINEERING DRAWINGS

In selecting a type of drawing projection, one of the major considerations is the *ease of measuring.* In assessing the value of the type of projection — orthographic, oblique, isometric, dimetric and trimetric views as well as true perspectives — we realize that measuring becomes more difficult when moving toward a true perspective view.

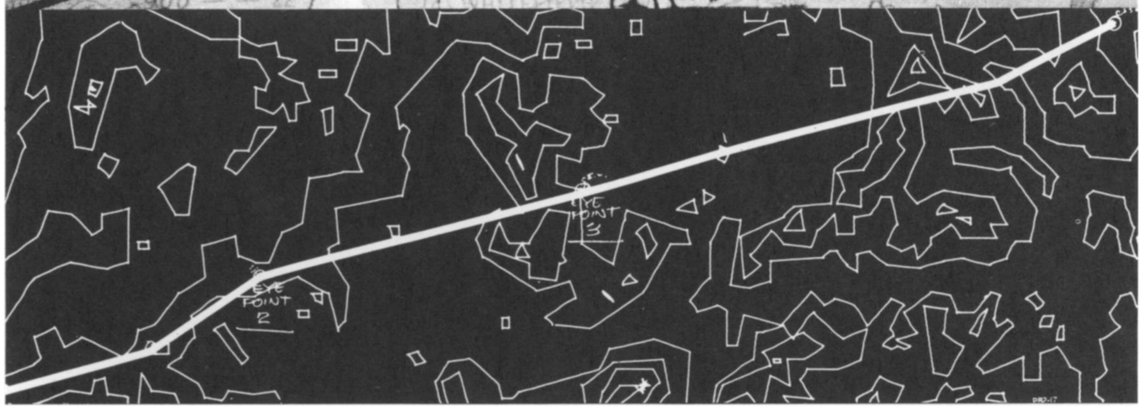
Another consideration in selecting a type of drawing projection is the *ease of understanding.* Since perspective projections represent objects just as the eye sees them, *measuring* becomes difficult while *understanding* increases. With the addition of shading, color, and especially motion to the perspective projection, the graphic effect can be most effective and thus facilitate communication.



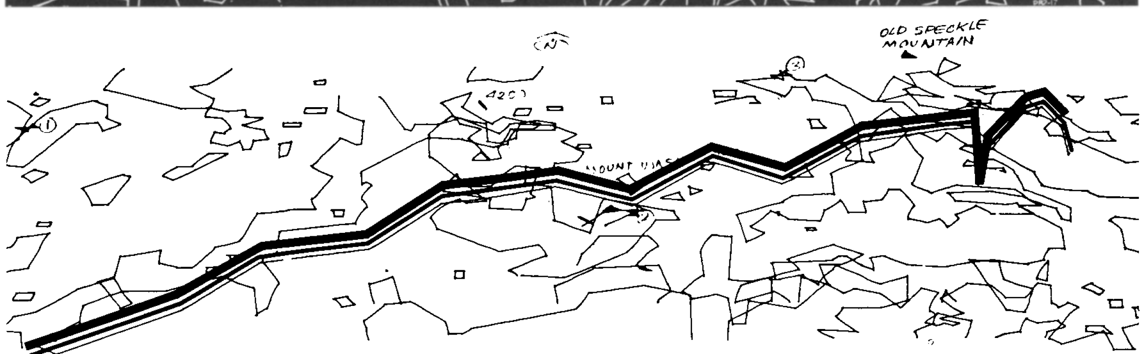
Perspective drawing of an airplane, constructed by a computer and delineated by a plotter. Any projection type can be generated after the information about the plane has been punched on IBM cards.

Once the engineering data is stored in the computer, Computer Graphics can present any type of projection from orthographic to oblique, isometric, dimetric and trimetric, to a true or central perspective. Any required combination of measurability and understanding plus details and parts of the projection can be selected and presented.

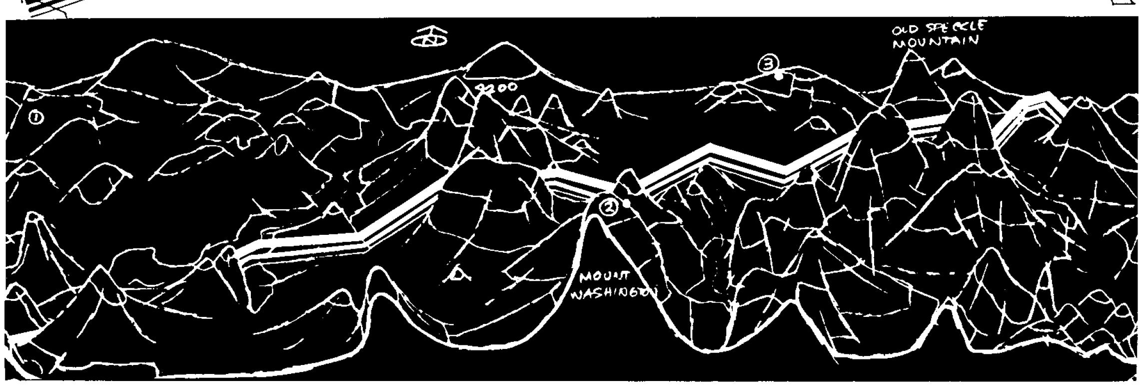
Original map with three hypothetical radar station locations.



Transcription of the above map, with simplified contour lines of the same terrain.



Computer drawn plot showing the area in perspective as seen from an altitude of 20,000 ft.

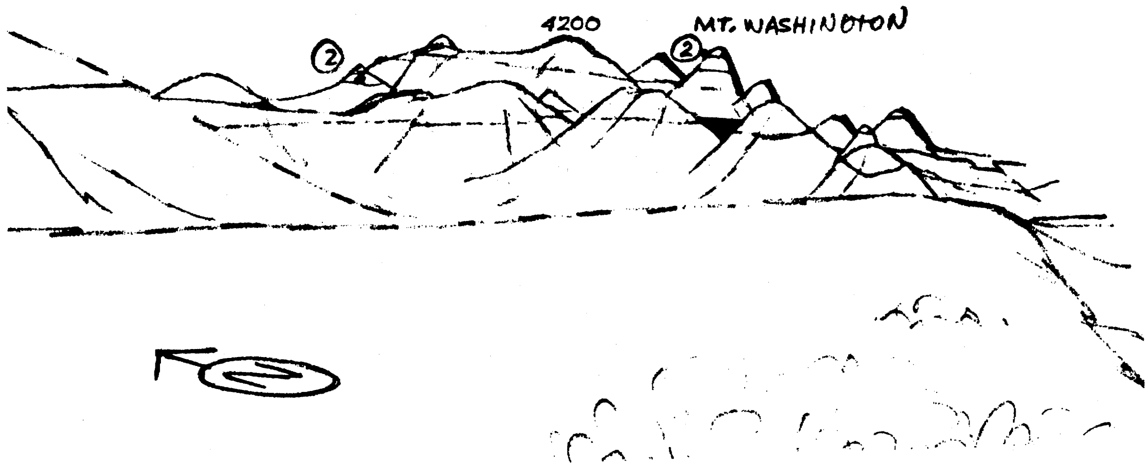


Rendered version of computer plot with hidden lines removed and mountain slopes drawn in.

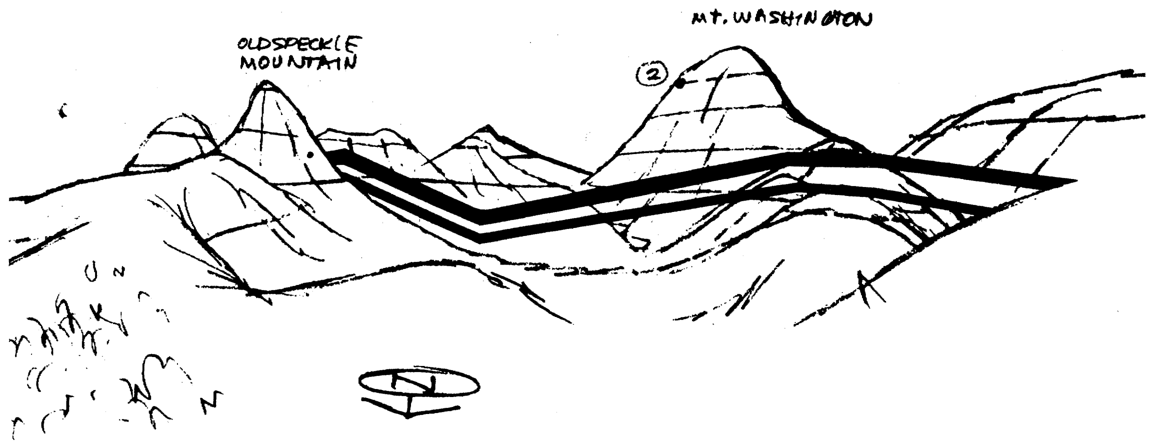
OPERATIONS ANALYSIS

Computer Graphics was also applied to Operations Analysis in a study concerning an area of mountainous terrain in New England. The object was to determine the location of three radar stations by tracking an aircraft flying along a selected flight path at two different altitudes, to plot the terrain interference and to measure the exposure of the flight path to each radar station at 500 and 200 foot ground clearances. A minimum number of points on a contour map was selected for economy in proving the method, and an orthographic *computer plot* was obtained in the exact size of the original map. To assure accuracy this plot was checked by overlaying it on the map.

The computer was then provided with a program that established a viewing point at an altitude of 20,000 feet and six miles from the center of the area. A plotter drawing of an



View from one radar station in which only a small amount of the flight path is visible (solid black area).



View of the same terrain from a different radar station showing a large segment of both flight paths.

over-all perspective was then obtained, showing the area as seen from the selected viewpoint, with the flight paths curving along the mountains and valleys. Working from this plot, the designer produced a more understandable picture by purposely exaggerating the vertical dimensions, drawing in the slopes and leaving in the visible contour lines. The hidden contour lines, also drawn by the computer, had to be removed manually. This conversion to a solid appearance was made with an accuracy of plus or minus 50 feet for each mile from the viewing point. Using a similar combination of computer and artist's drawings, views from each radar station were produced in order to determine where and for how long the airplane would be visible.

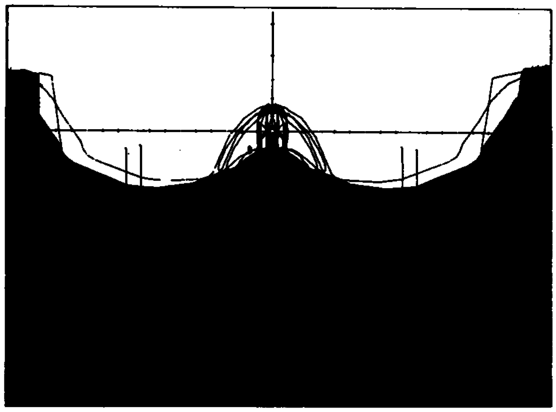
It was found that, in some views, tracking periods were very short — in other views, long segments of both the upper and lower flight paths could be observed at a particular radar station. These pictures, rather than mathematics alone, allowed trained radar personnel to see those parts of the flight path that would not appear on a radar screen because of its relationship to the terrain and the radar's characteristics. From these drawings it was then possible to make a map summary of the radar visibility of the airplane, at both altitudes, from any of the radar stations and thus evaluate the location of the radar stations.

HUMAN FACTORS APPLICATIONS

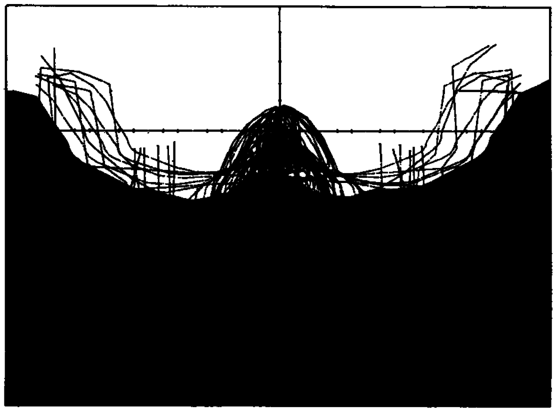
The need to know more about the visibility characteristics of a preliminary cockpit design prompted a study in the human factors area. The object of this particular study was to determine the cockpit design of a combat aircraft emphasizing visibility which is of vital importance for the aircraft.

A single plot of cockpit visibility is very tedious and costly if manually produced. During cockpit development constant refinements are necessary in order to achieve the best design. Computer Graphics using the previously described techniques, make it possible to have the results available in a form that is understood by the engineer and the non-technical observer as well. In the illustrated 360 degree view from a cockpit, the vision straight ahead is located at the center of the drawing, and the degrees of vision to the left and right are flattened out so that the left and right edges of the sheet actually represent the straight backward view. The horizontal line represents the full 360 degrees of the horizon, while the areas shown in black indicate blocked visibility.

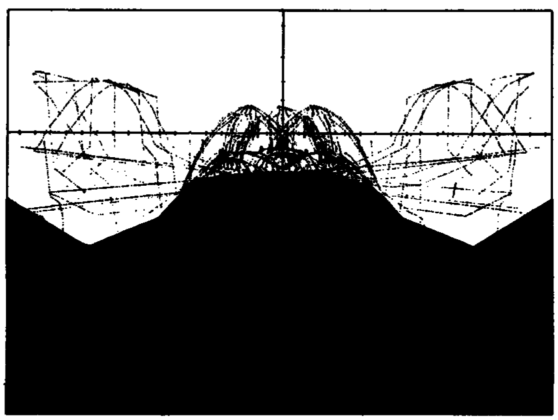
The visibility improvements which are obtained by turning and tilting the pilot's head were demonstrated by selecting a number of extreme viewpoints. All of these views were then automatically drawn on one sheet. The further improvement of the field of vision was shown for movements of the pilot's upper body.



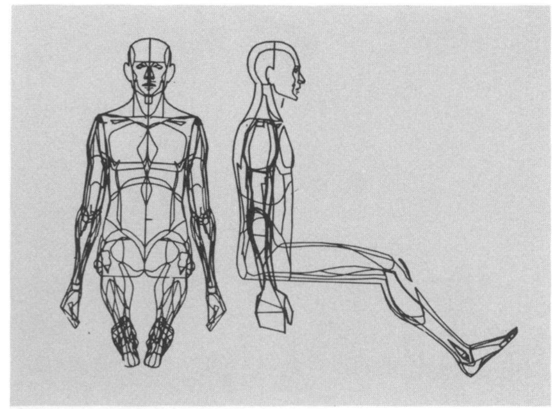
Computer plots of cockpit visibility studies for a combat aircraft. The black area indicates where visibility is blocked.

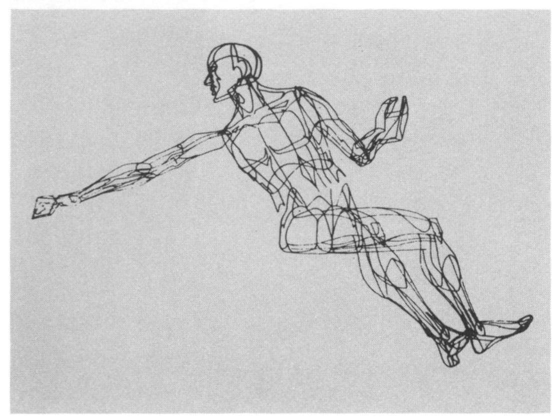


Visibility is increased by turning and tilting of the pilot's head.



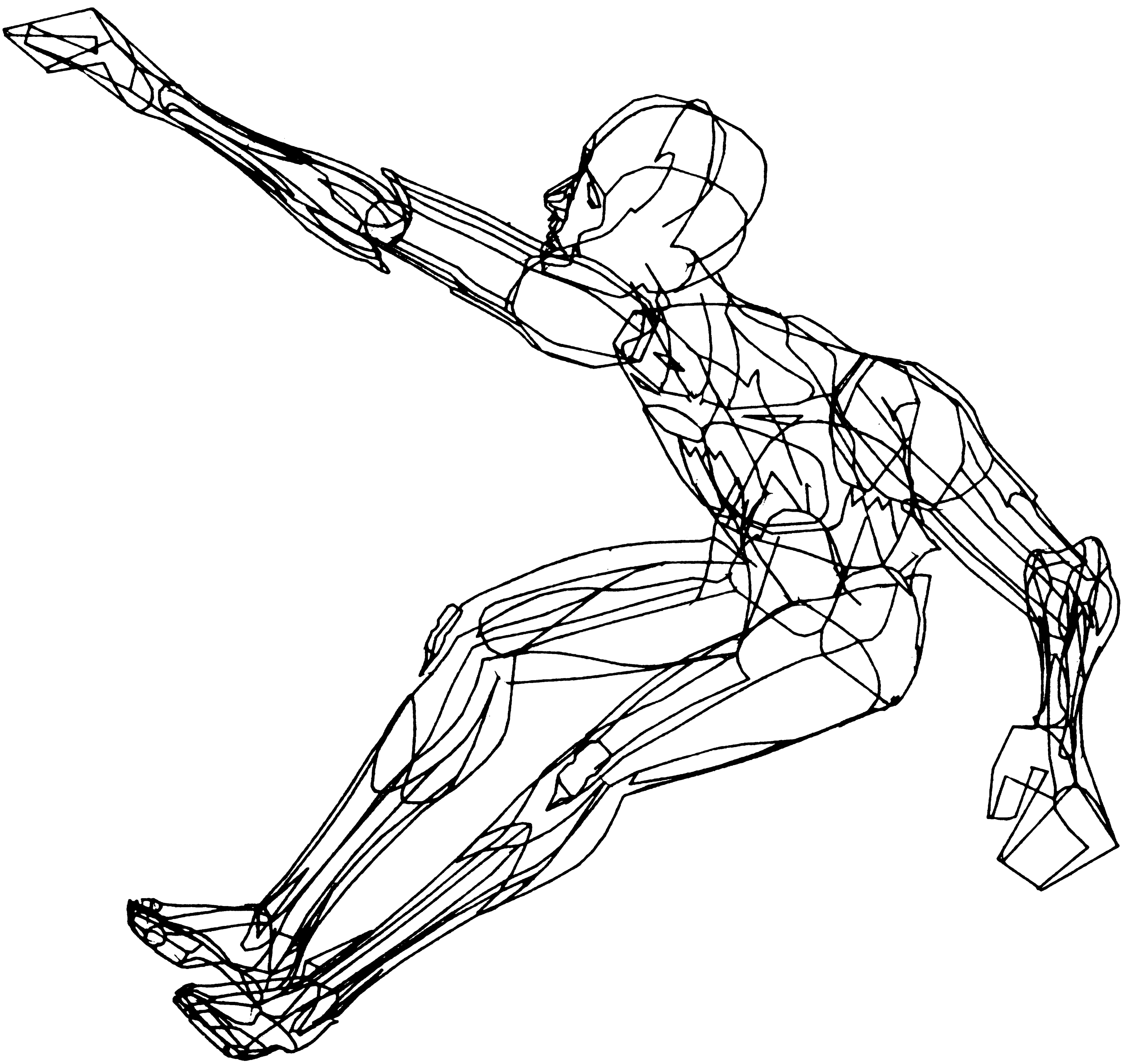
Visibility is further improved by including movement of the pilot's upper body.





Top right and center: Animated human figure designed, stored and drawn by the computer. This figure can be used to visually determine all types of reach situations, and is extremely useful in designs involving human factors.

One of the most exciting possibilities for Computer Graphics techniques is represented in an animated human figure, which has been designed, stored, and drawn by computer for personnel subsystems studies. The figure is accurately measured to meet Air Force anthropometric data for what is called a 50 percentile figure (the figure size of about 50 percent of Air Force pilots). The figure is composed of seven articulated systems and used for human factors studies. It can be easily animated for extreme reach situations as well as for plain sitting. Future figures used in Computer Graphics will have variable percentile ratings and will contain a larger number of systems for greater accuracy and realism of body motion.



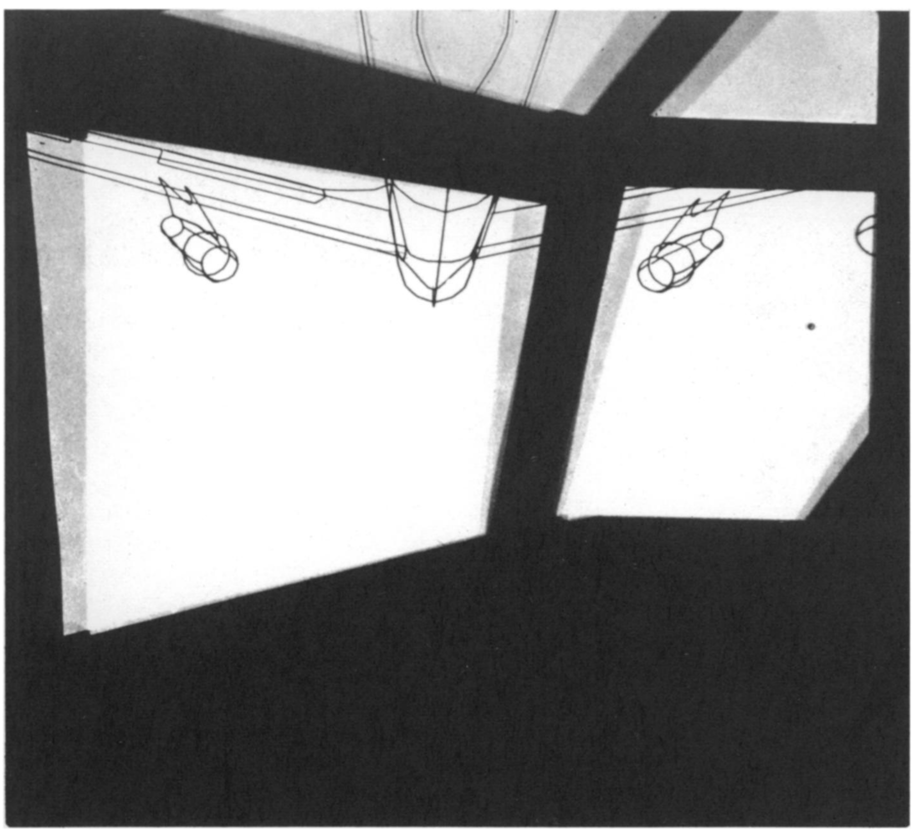
Animated human figure drawn by the computer and here shown in extreme reach situation.

Some Computer Graphics drawings may be used by the design engineer right at his desk to evaluate the suitability of a cockpit design quite early in the design effort. By overlaying a carrier drawing with a transparent sheet indicating the cockpit, the designer is able to examine effects on visibility during carrier landings from different airplane approach altitudes. The same technique can be used to simulate what would be seen of an existing airplane during refueling operations. The precisely computed images of the complex movements of the airplane maneuvering into position to be refueled in the air can be illustrated to show realistically an approach to the tanker up to the view at refueling contact. This simulation can be used as a basis for comparing cockpit designs for an airplane still on the drawing board.

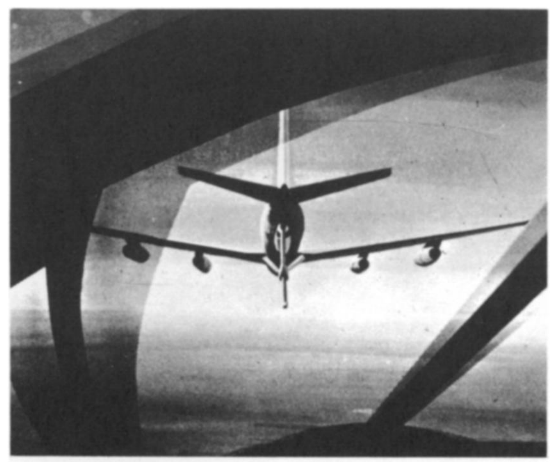
Computer Graphics is being applied to Supersonic Transport design especially in comparisons of preliminary nose configurations with those of existing aircraft. Renderings in color, based on plotter drawings and shown in quick succession, can be used to demonstrate cockpit designs which show the projected effect during a landing. Other Computer Graphics contributions to the Supersonic Transport program at Boeing involved production of a diorama for the evaluation of three different Supersonic Transport cabin and nose mockup configurations. A specific curvature of the diorama was calculated in order to be visually correct and the view from the mockup placed in the center was accurate for a precise location and altitude simulation in the final approach.

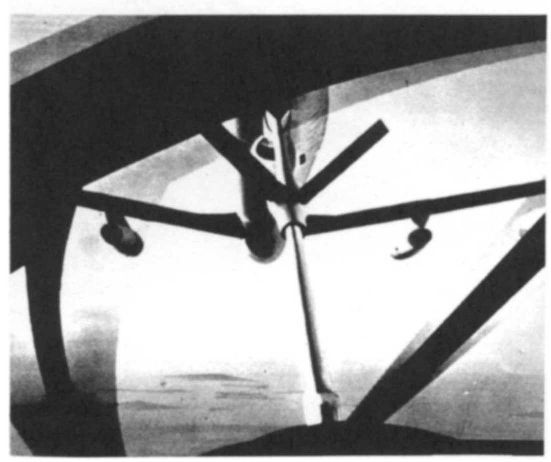
COCKPIT DISPLAY SYSTEMS

The complexity of instruments and devices necessary in today's aircraft cockpits is continually increasing. Design efforts in this area have to be directed toward presenting the information to the pilot in as understandable a form as possible. Because demonstration cockpit display designs are very costly and limited in effectiveness, Computer Graphics produced a film presentation in which a cockpit design was simulated in operation before the system itself was constructed. The system was designed to show the pilot desired views of part of the ground ahead of the airplane by means of a television screen in the cockpit. The television image was not a true perspective view but a carefully transformed image which made it possible to identify specific areas easily and inspect objects on the ground ahead at slow motion.

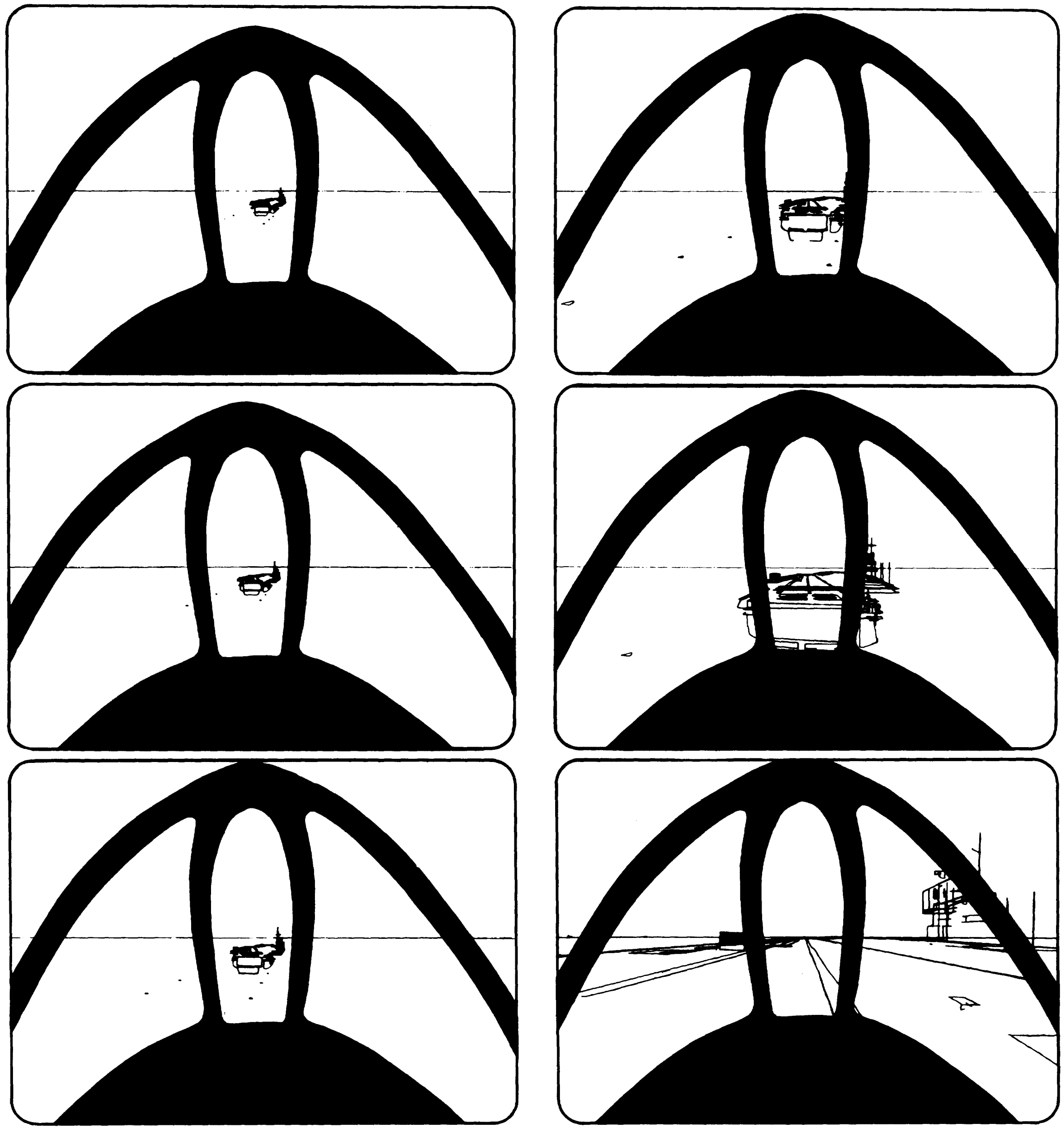


Top: Computer plot used as the basis for a rendering to demonstrate visibility in critical situation of mid-air refueling.





Top right and right: Renderings based on computer plots used for a Computer Graphics film showing cockpit visibility of an airplane yet to be built.



Frames of a Computer Graphics film on visibility studies during an aircraft carrier landing (black area is cockpit mask).

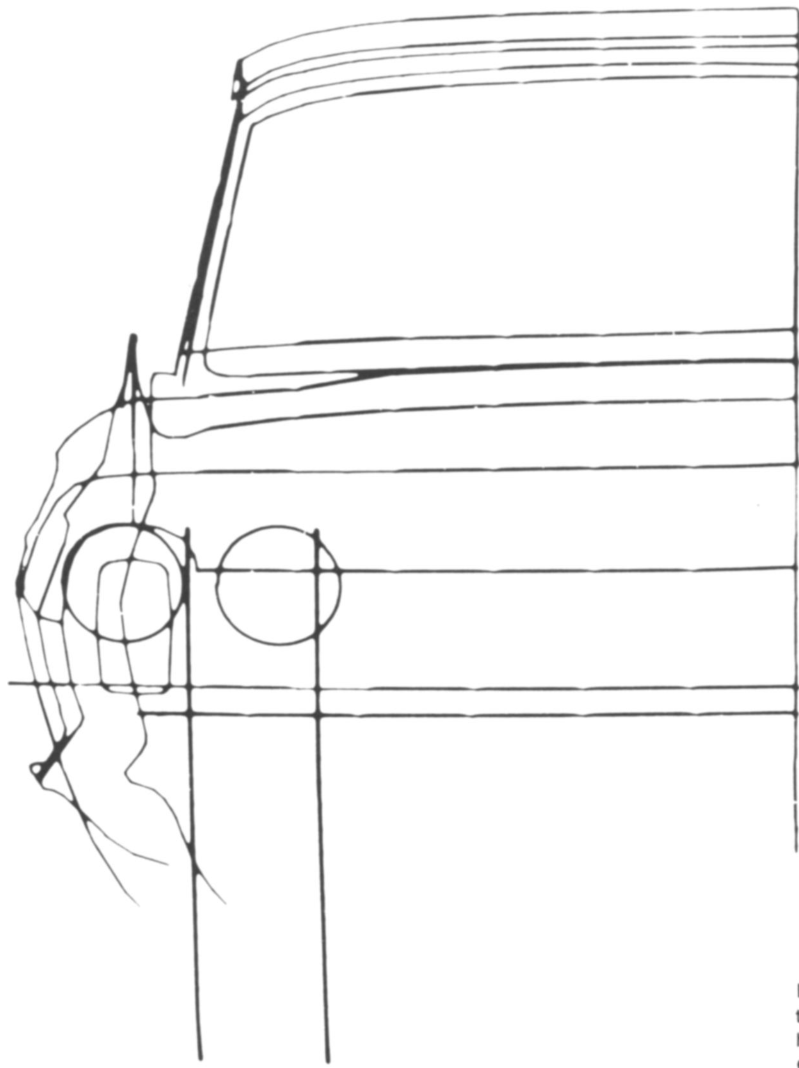
AIRCRAFT CARRIER LANDINGS

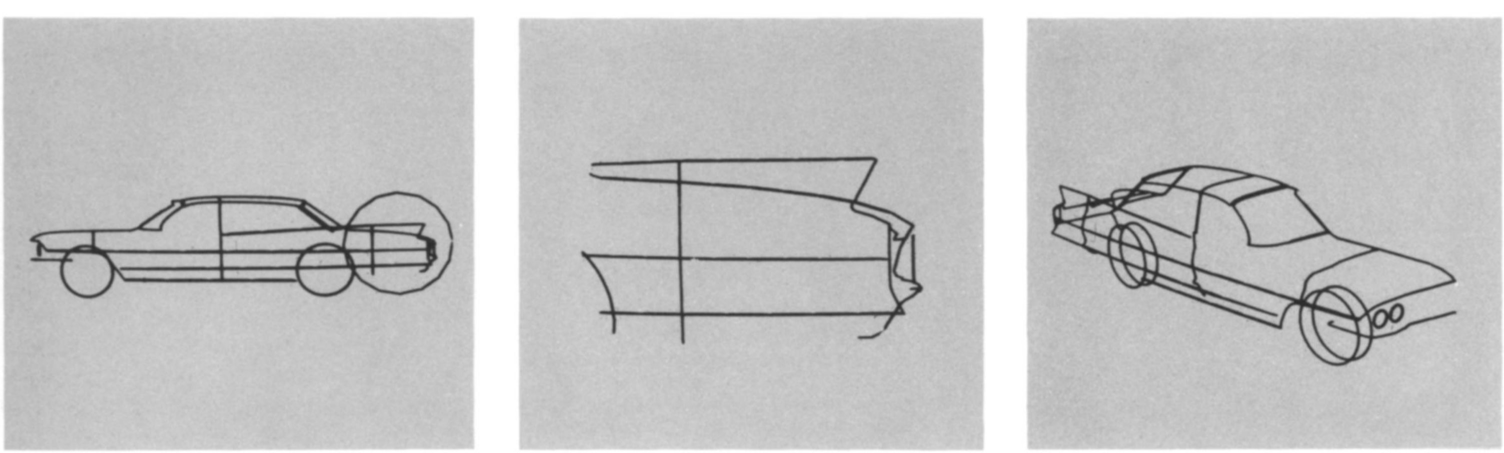
The difficult problem of testing the visibility of preliminary cockpit designs during a landing on a carrier can be best solved by producing a motion picture of the whole event. With Computer Graphics this film can be made before costly and time-consuming mockups are produced or actual landings are undertaken.

The landings can be demonstrated by computer plotting directly on standard animation cells and thus produce a motion picture that exactly simulates the pilot's view. The specific characteristics of the carrier had to be determined as well as airplane yaw, pitch and roll, speed and angle of descent. The basic animation, drawings at one second intervals, took everything into account, even the effects on the pilot's eye position after the compression of the shock absorbers when the airplane sets down.

Two hundred and forty computer drawn plots were needed to simulate the last ten seconds of a carrier landing. The animation drawings produced by Computer Graphics could then be turned over to the artist to add color and shading for an even more realistic motion picture.

Computer Graphics represents a new addition to the communication spectrum available to the designer. The almost unlimited capability of modern electronic data processing equipment, combined with the skill of a trained graphic designer, are the tools of this new communication medium. Effective, clear and pleasing visual solutions to engineering problems that have already been defined mathematically can be obtained through joint planning by engineers, graphic designers and computer programmers. Through thoughtful use of Computer Graphics, designers and engineers should have a more complete understanding of the problem at hand and should be able to produce designs more efficiently, economically and objectively.





Displays on the screen of the Graphic Console showing how the designer can call up enlargements of the drawing or different perspectives.

DESIGN AUGMENTED BY COMPUTERS

by Edwin L. Jacks

Edwin L. Jacks is Assistant Department Head of the Computer Technology Department at the General Motors Corporation, Warren, Mich.,where a computer system for the study of graphical man-machine communication was established.

Designers in the various automobile manufacturing plants work with blackboard sketches, large drawings, renderings, scale models and sculpture details in many different versions. Proposals are developed by the designer and analyzed by the engineer; adjustments are made and, finally, prototypes produced. Graphical techniques are basic to the design process because they serve as a mechanism for design evolution. Graphical communication between the designer and the engineer, therefore, plays a vital role in automobile design and production.

The computer system at General Motors enables the designer to produce, manipulate and evaluate free-form lines and surfaces without writing complicated program statements. Every element in the picture on the console screen is under the designer's control and can be changed instantaneously. He can revise and select elements on the screen and gradually develop any complex three-dimensional surface for an automobile.

The goal of such experiments is the development of a system where the designer can draw a rough sketch on the computer console screen, make changes and develop his ideas into a final and exact design without translating it into computer language.

Realizing the fundamental nature and importance of graphical communication techniques, the General Motors Research Laboratories has placed into operation an experimental facility for man-to-computer and computer-to-man graphic communication. The system, known as Design Augmented by Computers (DAC-1), provides:

a) a computer complex that permits rapid, high accuracy graphical input and output through a high-speed computer and

b) a flexible programming system to enable the convenient development of man-machine communication experiments.

Through this facility man-machine graphic communication is being explored as a potential technique for placing some of the designer's burden on the computer, thus giving him more time for truly creative work, and making it easier for him to design engineering changes. This could result in improved manufacturing processes, lower costs, and higher quality products.

Engineering has evolved rapidly during the past fifteen years as analysis techniques geared to the computational power of a slide rule and desk calculator have been replaced by techniques that make extensive use of computers. Graphical techniques for converting design ideas into final products, however, have not changed significantly, nor has the role of drawings in engineering design.

Engineering drawing plays a vital role in each phase of the evolution of a product. The original design proposals, the engineering analysis, the design compromises, and the prototype product fabrication all depend on graphical communication among engineers and designers. Whether the product is to be machined, assembled, stamped, wired, welded, or hand modeled, a drawing is made for review by the engineers concerned with the product. Prior to the final product drawing, many ideas are exchanged by means of sketches, drawings, plots, and engineering reports. In many mechanical design situations the two functions of engineering and preliminary product drafting are carried out by the same man. A drawing is his way of exploring design ideas.

Dependence on graphical techniques, therefore, is fundamental to the design process. Graphics serve as a language of communication among design personnel and as a mechanism for design evolution.

In 1952 the General Motors Research Laboratories began using a card-programmed digital computer for engineering and scientific analyses. These early studies showed that the computer could aid the design process. Notably absent from the applications, however, were problems related to graphical design. Subsequently, the G. M. Research Laboratories began a study on the potential role of computers in the graphical phases of design.

Prototype components and programs were developed to investigate the problems of image processing. A cathode ray tube recorder attached to a computer was already being used to plot results of engineering computations and satisfied the requirement for graphical output. An associated display unit provided a graphical display which, along with a simple switchboard, became an elementary man-machine console. A program-controlled film scanner was devised using the cathode ray tube recorder by substituting a photocell detector for the film magazine and connecting its output signal to a computer sense switch. With this setup, lines on film could be digitized under program control. Programs were written for graphic input and output and for the manipulation of images in three dimensions. These early programs and computer components were integrated into an operating system that demonstrated the feasibility of using the computer as an aid in the graphic design process.

On the basis of the early studies, the decision was made to establish a more comprehensive laboratory for graphical man-machine communication experiments. The facilities were to permit the computational power of a large-scale digital computer to be brought to bear on the problems of graphical design in a manner which

The DAC-1 system showing the Image Processor which provides input and output in graphical form (center) and the Graphic Console which contains a cathode ray tube, card reader, keyboard and program function switches and indicator lights (right).

would fully recognize the importance of man in design. This project has become known as DAC-1 (Design Augmented by Computer). (1-4)

The initial goal of the DAC-1 project was to develop a combination of computer hardware and software which would *permit "two way" man-machine graphical communication* and furthermore *provide a maximum programming flexibility* and *ease of use for experimentation.*

The hardware for the system consists of an IBM 7094 computer enlarged with extra disc and drum storage devices, an IBM 7960 special *image processing system* that allows the computer to read and generate drawings, and a *graphic console* equipped with a viewing screen which serves as a direct, two-way communication link between the designer and the computer. The special image-processing system was built by International Business Machines Corporation to specifications provided by the General Motors Research Laboratories. (2)

The supporting software for DAC-1, developed by the Computer Technology Department of the G. M. Research Laboratories, consists of more than ¾ million operating instructions. This makes DAC-1 one of the largest programmed systems in the United States. The software features a multiprogramming system, an algebraic compiler,* a data channel command compiler, a dynamic storage assignment procedure, and extensive facilities for the storage, retrieval, and editing of programs and stored data.

Each major portion of the DAC-1 system — the special image processor, the computer with attached disc memory, the multiprogramming system, the monitoring system, the programming languages used for system development, and the disc filing programs — has contributed to the system's flexibility and ease of use for experimentation.

*Compiler: Systems which translate a problem into special command codes meaningful to the computer without the programmer actually preparing a machine language.

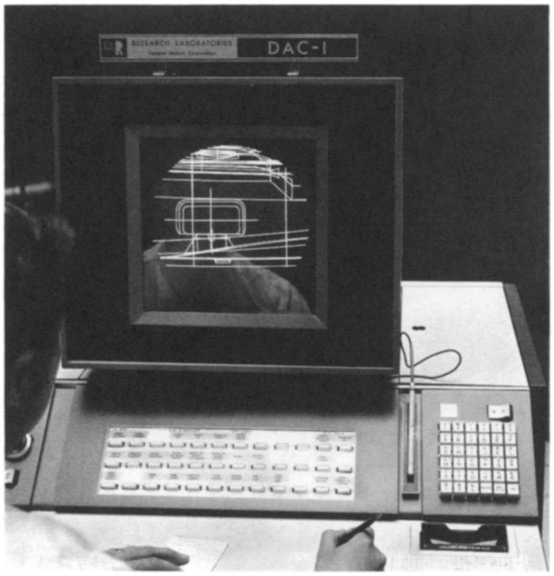
HOW THE DAC-1 IS USED

Before discussing the design objectives of the DAC-1 hardware and software, it would be valuable to describe how the system is utilized by the designer. First, using sketches and drawings as guides, the designer writes out statements of his problem — usually a descriptive geometry problem — in a design language. These statements are key punched on cards and loaded into the computer memory.

Once the group of statements is stored in the computer, the designer commands the computer to perform the functions requested in his program through the man-computer console. As the program is executed, the computer checks each statement for syntax and punctuation. If an error is detected, the computer stops and displays an error comment. Using the console equipment, the designer can inspect the statement, make corrections, and instruct the computer to proceed.

Upon completion of the task, the computer response usually will be visual display of lines forming a simple drawing. The designer studies the display at the graphic console to decide if it is an acceptable solution to the problem. He may enlarge any portion of the display or view it from any angle or perspective. If is is not satisfactory, he can modify it immediately at the console by adding or deleting lines, changing statement parameters, or entering alternate statements. If the display is acceptable, he can direct the computer to produce a permanent copy of the drawing using the image processor. After perhaps many changes, he may have DAC-1 produce control tapes for automatic drafting machines or machine tools.

Since the program itself is general, with details supplied at the console, it can be stored in a library of design programs. As this library grows, designers can select from it programs for those design functions they might require for a specific job.



The Graphic Console is the system's control point. Information displayed on the console screen, for example a detail of a car, can be modified through the position-indicating pencil, control keys or card-input.

IMAGE PROCESSING AND COMMUNICATION

The overall objective of the image processing system[2] was to achieve the equivalent of what is possible with graphical man-to-man communication while using drawings. In establishing systems specifications, four kinds of man-machine communication were sought: static, dynamic, comparison, and non-graphic.

Static Drawing: The objective was to produce a hard copy drawing for engineering use. Conversely, accept a drawing and, under computer control, read the drawing.

Because of the nature of automobile design, it was necessary that the DAC-1 system accept free form curves — curves which are constructed without consideration of particular mathematical representations. Furthermore, to provide compatibility with existing design procedures, precision input and output of such curves were needed. The drawing input-output functions are achieved in the image processor of the IBM 7960 system.

Dynamic Drawing: The objective was to simulate the type of man-to-man communication where one man draws or points at a particular part of a drawing while another man observes or discusses details of the design with the first man.

This capability is provided in the graphic console of the IBM 7960 computer through the combination of a 17-inch display tube and a device called a *position-indicating pencil.*

Comparison Function: The objective was to provide for the overlay of two pictures to permit comparisons of differences and similarities in the information.

This feature is provided by having the image processor designed so that pictures can be recorded on two separate film trains, then projected automatically onto a common view screen. This feature allows, for example, overlay of scanned data with the original film source for verification. By programming techniques, the graphic console also can be used to compare drawing information.

Non-graphic Information: The objective was to provide, via the graphic console, a convenient way to communicate alphabetic and numeric information to the computer, multiple choice decision responses to the computer, and permissible actions by the man.

For alphanumeric information, a 36-position keyboard with upper and lower case and a slow-speed card reader are part of the graphic console. For communication of gross actions such as review or transform, the console has 36 program control keys and 36 message lights. The computer receives a signal when a program control key is depressed by the man, and the man receives a visual signal when the computer turns on a message light.

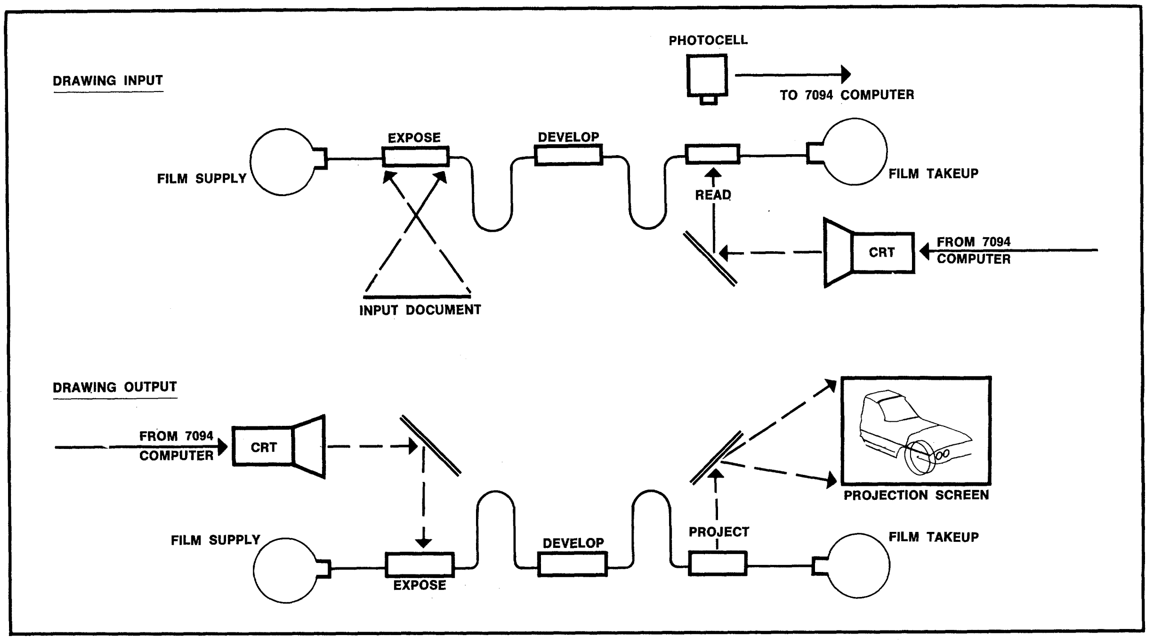
Studies at the General Motors Research Laboratories were made to estimate the computing facilities required to support adequately the DAC-1 project. Considered in the studies were the number of instructions required to support the experiments, execution time for the required programs, and man-machine response rate. These studies indicated that approximately 200,000 to 500,000 instructions would be programmed for the graphic communication experiments. The computation required for these experiments was estimated in terms of Central Processing Unit (CUP) use per hour and amounted to only six minutes of CUP time for each hour of DAC-1 console use.

The response rate considerations were stated in terms of system objectives. The designer essentially had to be working on-line with the computer and in *real time.** The measure of real time was that the man and machine could carry on a meaningful conversation about a design at the rate satisfactory to the man. The response consideration then required a real-time approach to receiving and handling data arriving from the man. But the computer programs and the hardware did not need to have a fail-safe time limit approach to sending a response to the man.

Another more independent consideration was the computing requirements of the General Motors Research Laboratories. In the late 1950s, an IBM 704 computer was in use between two and three shifts per day. It was forecasted that an IBM 7090 or an equivalent computer would be required by 1961 to satisfy the continuing needs for an engineering and scientific computing facility.

---

*Real time — the time the computer needs to respond with a solution.



Schematic diagram of the Image Processor. The Image Processor Unit contains cameras, projectors and cathode ray tubes (CRT) which enable the system to "read" and produce engineering drawings. Drawings are inserted in the unit via a drawer, and photographed onto 35 mm film. The film is then developed and positioned beneath a cathode ray tube and a photocell which are used to "read" lines from the document, convert them into digital language, and store them in the computer memory.

The Image Processor also can produce drawings on 35 mm film. The film is exposed to a cathode ray tube, developed, and projected on a viewing screen. Exposed film may then be mounted in cards for storage. Conventional reproduction copies can be made from the film to produce work drawings.

It was believed that an IBM 7090 computer* would meet the combined requirements of the DAC-1 system and other Research needs. The speed of the IBM 7090 would adequately handle the computational load and the machine would be able to give the response time desired for console comunication and computational purposes. To prevent wasting the estimated 54 minutes per hour of non-Central Processing Unit use by the man, multiprogramming techniques were to be developed and the computer would need to be adapted so that two independent programs could reside in its core memory with a minimum risk of either program modifying the other program. For this purpose, a core memory protection system was designed which prevents instructions from storing into program-specified 4096 word blocks of the memory.

Multiprogramming also implied that a clock should be attached to the computer to provide proper timekeeping for accounting purposes during the switching from program to program. A clock was built by the Delco Radio Division of General Motors with a millisecond as its basic interval of time.

The DAC-1 system requirements for 5×10^5 words of program storage could be satisfied by having a disc memory on the computer. The original 7090 configuration had a 1405 disc connected to the computer via a 1401 computer and a direct data connection. The present facility uses a 1301 disc and three drum storage units for the program and data library.

PROGRAMMING SYSTEMS

The combination of the IBM 7090 computer and the IBM 7960 special image-processing system provided an experimental graphical communication facility. To support this system from the software standpoint, programming techniques were developed to minimize the time from the conception of a man-machine communication experiment until the required programs were operating.

The programs had to be able to display situations to the man conveniently. If the man was expected to require more than a millisecond to respond, the programs had to be able to say to a control program, "Control, I am in standby status now and when the man answers my question or takes other action, return control to me."

For programming convenience, the programmer had to be able to do all his programming in a higher level language (higher than an assembly type language, at least) including the programming of the data channel driving the special image-processing system, the loading of programs by name from the disc, and the analysis of all data coming from the image processing or graphic console equipment. In short, he had to be able to program all of his graphical communication experiments in a language similar to FORTRAN* or ALGOL.* An algebraic compiler and a data channel command compiler were developed for this purpose.

The specifications of the programming system revolved around three broad statements of facility operational policy. First, for programming purposes, the 32,768 word computer memory was to be considered as two blocks, with half of it assigned to the DAC-1 console support programs and the other half assigned to the stan-

*In 1963, the originally installed 7090 was upgraded to a 7094.

*Fortran is a composite of FORmula TRANslation, a language that converts programs written in mathematical symbols into machine language.

*ALGOL or ALGorithmic Oriented Language, is an international procedure-oriented language.

dard batch monitor operation. Secondly, all input/output programs in both the DAC-1 and batch monitor operations must use the trapping hardware built onto the Central Processing Unit, and all trap program operations must be compatible with a Trap Control System (TCS) developed by the General Motors Research Laboratories. The Trap Control System program prevents chaos when all five channels simultaneously request priority activity from the Central Processing Unit. The trapping hardware allows each of the five data channels to direct the Central Processing Unit to stop its current activity and handle a priority job for the data channel.

The third policy statement indicated that the batch monitor's use of the computer was limited to the two channels for the tape units while the DAC-1 console program's use was limited to the channels connected to the Central Processing Unit and the disc and drum storage units. This condition was imposed to prevent conflicts in hardware use and means. For example, tapes in use by the batch monitor could not be used by DAC-1 during multiprogramming. One major exception to this rule was that use of the disc was permitted by programs being executed under batch monitor control for purposes of compiling or checking out programs being developed as part of the DAC-1 project.

MAN-MACHINE CONCEPT

The DAC-1 system has been in operation eight hours per day, utilizing extensively the systems hardware and software capabilities discussed here. From the standpoint of a laboratory facility, system performance is excellent. It has proven that man and machine can communicate readily via graphical means.

With the combination of computer-controlled image scanning and man-machine communication,[4] a simple computer program rapidly solves the problem of conversion from graphics to binary data. When any uncertainties arise, the man, as referee, can obtain control. One example of an uncertainty is low contrast input film, which is difficult to scan.

It could be argued that for each uncertainty a program could be written to analyze the situation, then man would not be needed to aid the process. The strong point of man-machine communication via graphic consoles, however, is that for any given problem one may ask which parts are easily solved by the computer and which are best solved by man. This results in programs being written which have decision points in them at which the man at the console can be asked for advice.

Many of the past discussions of man-machine communication have been based on the concept of "let the man get to the computer" so he can ask questions directly of the computer program. Experience at General Motors Research to date has been that that payoff comes not from asking the computer a question, but from assigning the computer a task from which the response is one of the following: "What is my next job? Here is the answer; what next?" or, "I don't understand, but here is my analysis of the situation."

SUMMARY

The hardware for the DAC-1 system includes a large computer augmented with extra disc and drum storage devices, a special image processor that allows the computer to read and produce drawings, and a graphic console that serves as a direct, two-way communications link between the man and the computer.

The software incorporates three major departures from conventional higher level language programming: multiprogramming, source program storage allocation control, and a disc library of programs that is available during program execution.

Multiprogramming permits computer programs to achieve efficient use of the processing unit even though they work at the man's pace. Program storage allocation control allows each program to adjust storage assignment dynamically as a function of data needs. The disc library available at execution allows a control subroutine to view other subroutines as units which require certain inputs to produce outputs. This feature allows continued growth of the design support programs with no change to control programs.

With the new laboratory facilities at the General Motors Research Laboratories, the process of man-machine communication for design is being explored now with both formal (direct comparisons of methods with planned testing) and informal (trying something to see how it works) experiments.

REFERENCES

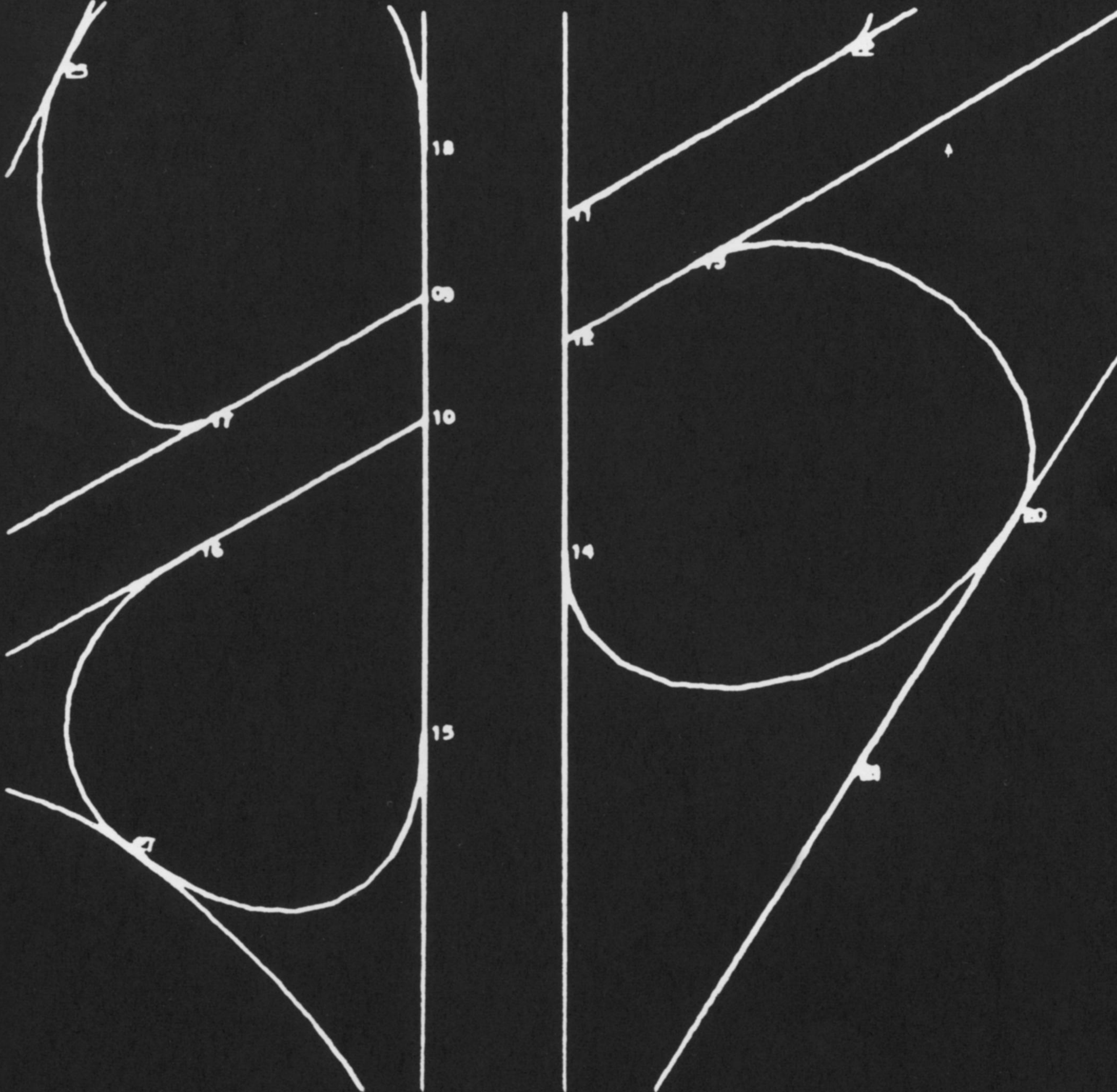
1. Cole, M. P., Dorn, P. H., and Lewis, C. R., "Operational Software in a Disc Oriented System," *Proceedings of the Fall Joint Computer Conference,* San Francisco, October 27-29, 1964

2. Hargreaves, B., Joyce, J. C., Cole, G. L., et al, "Image Processing Hardware for a Man-Machine Graphical Communication System," *Proceedings of the Fall Joint Computer Conference,* San Francisco, October 27-29, 1964

3. Allen, T. R. and Foote, J. E., "Input/Output Software Capability and Image Processing System," *Proceedings of the Fall Joint Computer Conference,* San Francisco, October 27-29, 1964

4. Krull, F. N. and Foote, J. E., "A Line Scanning System Controlled from an On-Line Console," *Proceedings of the Fall Joint Computer Conference,* San Francisco, October 27-29, 1964

Highway interchange photographed on the screen of a cathode ray tube. The designer can modify the design and recall an enlargement to examine or work on a detail.



Dr. Marvin L. Manheim is assistant professor in the department of Civil Engineering at the Massachusetts Institute of Technology, Cambridge, Mass. He is particularly interested in the role of the computer in design and has published papers and reports concerning this subject. Together with Dr. Christopher Alexander, Prof. Manheim worked on research projects in the area of highway route location and the design of highway interchanges.

The Problem-Solving Process as presented here should be understood as a system for solving complex design problems with the use of computers. Projects like the route location of a new highway and problems in architecture or city planning are of such complexity that the computer should be employed as a tool for analysis and prediction. The Problem-Solving Process is a sequence of procedures with emphasis on Search and Selection. To achieve the most correct action or plan within a complex situation, different methods are used to produce alternatives and then a preference ordering of these actions is established. The most desirable plan is selected and implemented. The designer decides what parts of the procedures the computer should be used for and retains the prerogative to find alternative solutions with his own creative talent.

Properly designed highways, hospitals, bridges, etc. are urgently needed. Therefore, the most efficient means should be used to find solutions within the given restraints of time, costs and manpower. The trial and error possibilities inherent in this Problem-Solving Process are most valuable since they make it possible to test, re-evaluate and revise any design situation. This is extremely important if we realize that a highway, building or bridge once designed and built cannot be removed as one would withdraw a badly designed product from the market.

PROBLEM-SOLVING PROCESSES IN PLANNING AND DESIGN

by Marvin L. Manheim

In design and planning, problems of increasing complexity have to be solved today. Computers and elaborate programming techniques can now assist the designer in finding solutions to highly complex problems. To utilize these technologies effectively, it is necessary to better understand how man and machine can work together successfully in attacking complex problems. A general theory of Problem-Solving Processes, particularly applicable to planning and design, is needed.

The necessity for such a theory is emphasized by some very exciting work being done in computer graphics — the various ways in which information and images generated by the computers can be displayed and operated upon. But the availability of such techniques alone will not enable the designer to utilize these accomplishments in the planning and design process. He must be able to answer such fundamental questions as what information should be displayed, when, how, and at what points in the process.

The objective then is to formulate a theory which can be implemented and useful. The Department of Civil Engineering at the Massachusetts Institute of Technology is developing an interrelated system of computer programs for addressing a broad spectrum of design and planning problems in civil engineering, soils, structures, water resources, transportation and other areas. This system is called ICES (Integrated Civil Engineering System). Hopefully, many of the following ideas will be implemented in this system and subsequently tried out in applications to transportation and other engineering, planning and design problems.

Apart from this pragmatic objective, the ideas presented here are still exploratory and tentative. They are biased toward transportation planning, city planning and economic planning; but it is hoped that they will also apply to design, especially architecture, and the general problems of socio-political planning.

A "Problem-Solving Process" is defined as any man-machine system interacting with a problem in order to develop, select, implement, monitor and revise actions in the real world. The characteristics of this Problem-Solving Process should be identified so completely that the model is applicable to problems in architecture, design, transportation, urban and economic planning.

The Problem-Solving Process is visualized as containing a variety of procedures to be used in the Problem-Solving Process when and as appropriate. The emphasis is on the concept of a "process": a sequence of many different kinds of activities.

The basic frame of reference is the new computer hardware and software which can provide, by use of the "time-sharing" systems,[1] a flexible, highly-interactive service to many designers through remote-access consoles.

An example from the area of transportation planning should demonstrate the need for thinking in terms of a <u>total</u> Problem-Solving Process and show just <u>how</u> complex real problems can be.

The basic characteristics of a transportation planning problem are reflected in:

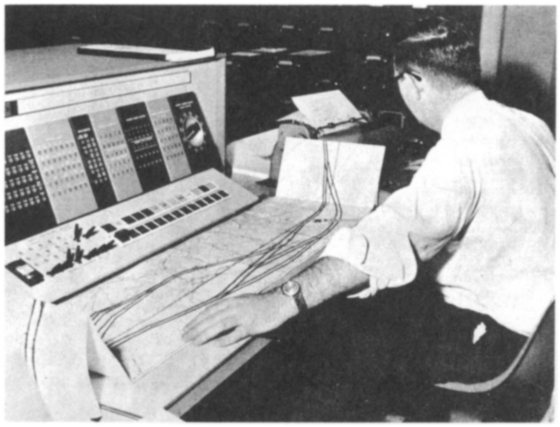
a) <u>The variety of options open:</u> In transportation planning the alternative options (facilities, vehicle modes, operating policies, etc.) are not well defined, and cannot be easily generated or enumerated. They are too complex to describe and very difficult to manipulate.[2]

b) <u>The variety of impacts which must be considered:</u> The impacts of transportation planning such as construction and maintenance costs, tax revenues, population displacement and changes in social structure are also numerous and not well defined. Evaluating the full set of impacts for each alternative plan is difficult as many desirable and undesirable effects must be balanced, and no single "quantitative measure of effectiveness" is available to summarize the issues.

c) The number and complexity of interactions of the different transportation models required to predict the impacts of a specific plan: Predicting such impacts is computationally very difficult, relatively expensive and, in the end result, still uncertain.

Because of the *variety and complexity of the options,* the space of alternative plans is not well structured, which makes the production of desirable alternatives very difficult. Because of the *variety of impacts,* no *single* criterion function can be optimized. Because of the number and complexity of the transportation models which are available to the designer, determining the desirability of a new alternative is also difficult. For these reasons, a single optimizing model or plan can never fully solve such a transportation problem.

Transportation planning is an example of the very large class of problems Walter Reitman calls "ill-defined,"[3] in contrast to Minsky's definition of "well-defined" problems.[4] The characteristics which are identified here apply as well to urban planning problems and architecture as to other areas of planning and design.



Computer-oriented systems are being adapted to the language of the highway engineer to evaluate highway location and design decisions.

Plotter attached to an IBM data processing system automatically plotting highway location designs.

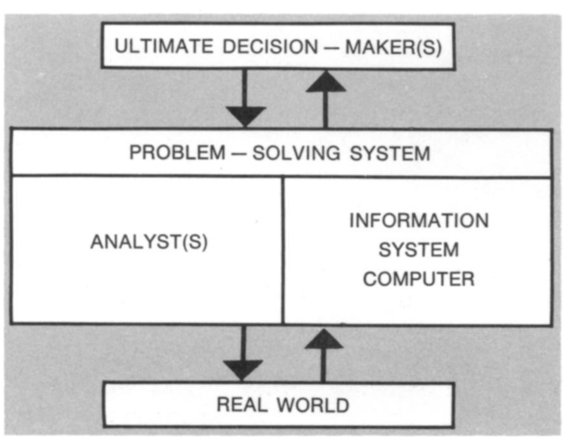
MODEL OF A PROBLEM-SOLVING PROCESS

Data and procedures: The basic elements of a Problem-Solving Process are divided into two major categories parallel to common computer usage: Data and Procedures. The *Data* is what might be represented as "files" in a computer system and includes everything from action to goals. The *Procedures* or "routines," which include procedures for Search, Prediction, Evaluation and Choice, operate upon the files to produce other files.

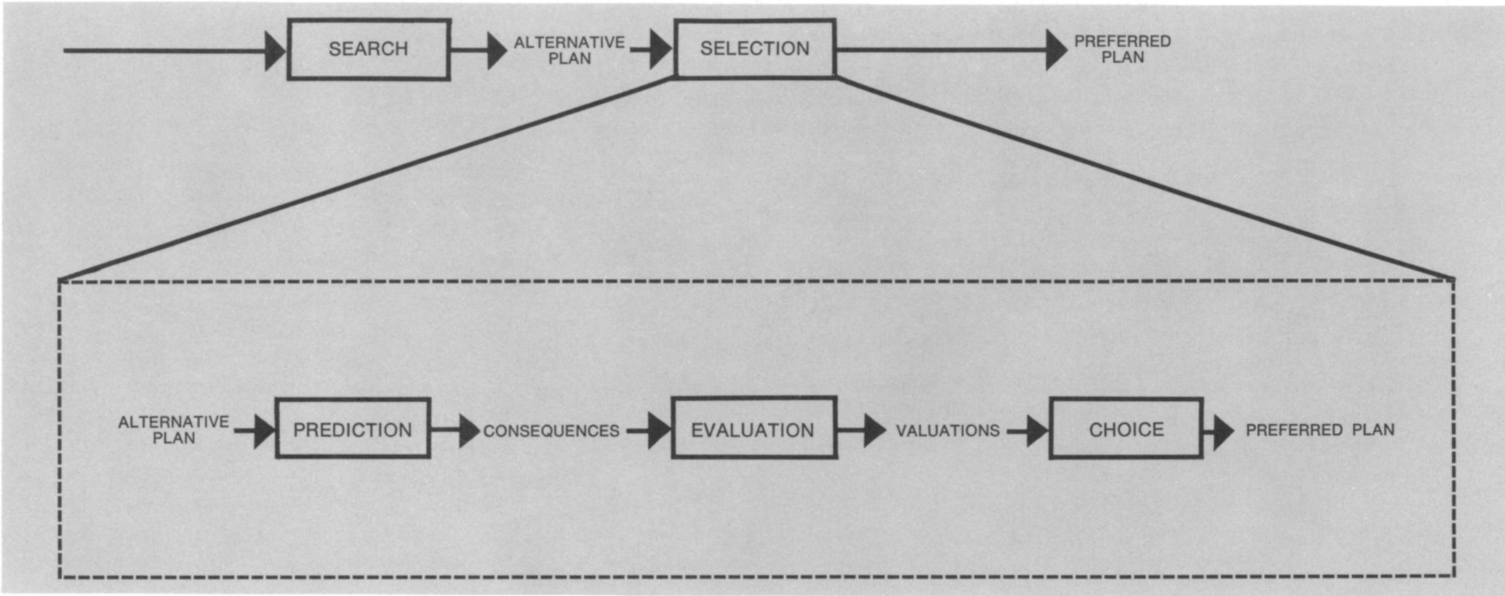
BASIC PROBLEM-SOLVING PROCEDURES

The basic focus of the Problem-Solving Process is on actions which potentially can be implemented in the real world. A Problem-Solving Process concerns the development, selection, implementation, monitoring and revision of actions. The actions may concern building designs, transportation plans, or even economic policies.

The basic view of the Problem-Solving Process is as follows: Alternative plans or actions are produced, and then a preference ordering over those alternatives is established. If the most desirable alternative is sufficiently attractive, then the Problem-Solving Process ceases and that most desirable action is implemented in the real world. However, if this plan is not sufficiently desirable or attractive, then Search



A Problem-Solving Process consists of an analyst (the designer or engineer) and the computer system, interacting between the ultimate decision-maker and the real world.



Basic Problem-Solving Module: The essence of the Problem-Solving Process is that alternative plans of action generated during a Search phase are selected, the consequences of these alternative plans are predicted, evaluated and a choice of the preferred alternative is made.

is repeated and new actions are generated. The sequence is repeated again and again, until finally there is one action sufficiently attractive for implementation in the real world.

This image of a "trial and error" process is basic to this concept of a Problem-Solving Process. Of course, it is completely contrary to the image of a problem for which the optimal solution is obtained directly by "solving" a mathematical model. Such "optimizing" methods correspond to one Search and Selection sequence only and do have an important role in the broader Problem-Solving Process, but as the transportation planning example indicates, real problems are too complex for such techniques to carry the whole burden.

The basic activity of the Problem-Solving Process in any design problem is to produce actions and then choose among them for a solution. The following Search and Selection procedures perform these functions:

Search designates any procedure used to produce one or more alternative plans or actions. Search may be intuitive, as in the sense of "design," or may be formalized, as in a mathematically-formulated problem.

Selection designates the process of choosing among several alternative actions. The input to Selection is a set of alternative actions. The output of Selection is a "preference ordering," or ranking of the actions by desirability. To actually accomplish Selection, three basically different kinds of procedures are required:

a) Prediction: Procedures for Prediction are used to anticipate the consequences which an action would have if implemented in the real world — for example, to predict the reliability and weight of materials in a particular structural design.

b) Evaluation: Procedures for Evaluation operate upon the predicted consequences to yield statements of the valuations, or relative desirabilities of those consequences — for example, the degree to which the structural reliability is satisfactory. All predicted consequences cannot adequately be represented by a single measure of value. For example, costs, safety and aesthetics cannot all be lumped into a single measure of value such as dollars, or some other overall utility measure.

c) Choice: Because of the absence of such a single measure, Evaluation must be followed by Choice. In Choice, each action is compared on the basis of its set of valuations — cost, safety, aesthetic quality, etc. — and then a decision is made about the rankings or preference ordering of the actions.

DATA FILES ASSOCIATED WITH THE BASIC PROBLEM-SOLVING PROCEDURES

There are several types of Data associated with Search and Selection which are filed in the appropriate data files. Each time Search produces new actions, these are added to the data files for action. The current statements of goals relevant to the Problem-Solving Process are stored in the appropriate files. Selection produces data about the consequences of each action, and the valuations of those consequences. The output of Selection is stored in the file containing the latest ranking over the actions.

Search and Selection procedures are used many times in a Problem-Solving Process. Each use results in changes and/or additions to the files for currently open actions, consequences of actions, valuations of those consequences and ranking of actions. To a lesser extent, changes will also occur in the files for goals.

ADDITIONAL PROBLEM-SOLVING PROCEDURES

Because Search and Selection procedures concern the basic generation and selection of actions, these procedures are at the heart of the Problem-Solving Process. But there are a variety of other activities which must occur in a Problem-Solving Process to allow Search and Selection to operate, and to revise the context in which they operate.

Goals are set initially, but may change radically during the operation of the Problem-Solving Process. Consequently, Goal Formulation and revision procedures play an important role.

Information about the state of the real world is continually flowing into the Problem-Solving Process: Information analysis procedures edit and organize this raw data, adding the raw observations to the data base, eliminating actions from the file of currently open actions which are no longer open, providing information to and triggering the goal revision procedures, and revising probability distributions which represent uncertainties. The variables subject to uncertainty are carried in special files, and the current probability distributions over the values of those variables are maintained in other files. These uncertainties are an explicit part of both Prediction and Evaluation and form part of the difficulty of decision in Choice, in that not only the conflicting valuations, but also the explicit uncertainties must be balanced.

Hypothesis generation as well as Calibration and Validation are special procedures which are required to develop the models or plans to be used in Prediction and, to a lesser extent, in Evaluation.

Decomposition and restructuring procedures are concerned with how a problem is broken into subproblems for greater ease in analysis, and how the overall solution is put together from partial solutions to subproblems. For example, in the project of locating a highway, the problem is identified in a list of requirements and a set of interactions which form a tree, a hierarchical listing of subproblems. The overall solution is achieved by first solving the subproblems on the lowest level of the tree and working upward to the top of the tree. For each subproblem, appropriate Search and Selection procedures are required.

Metaprocedures form the overall "executive" of the Problem-Solving Process. At any point in a Problem-Solving Process there are a large number of alternative procedures potentially useful, and a decision must be made as to which procedure to use next. Metaprocedures are used to make this choice. The procedure selected must recognize the fact that resources for problem solving are limited — time, cost and manpower, for instance, are typical constraints.

CONCLUSIONS

A number of principles are basic to this concept of a Problem-Solving Process. These principles should be viewed in their relation to the Problem-Solving Process model.

A Problem-Solving Process involves the application of a variety of different procedures. The basic procedures are Search, Prediction, Evaluation and Choice. The basic procedures of Search and Selection produce the actions, and are iterated many times. Supplemental procedures include Goal Formulation and Revision, Information Analysis, Hypothesis Generation, Decomposition and Restructuring Procedures and Metaprocedures. A full man-machine system for problem solving must possess and make use of all of these types of procedures.

The roles of man and machine will, in general, be different for each procedure in a Problem-Solving Process. A key theme underlying this discussion is the balance between human roles and machine roles in a Problem-Solving Process. The overall objective is to enhance creativity, not stifle it. As Serge Chermayeff put it in describing the myth of conflict between Rationality and Inspiration . . . "Rationality as a system of procedure does not exclude inspiration which acts as an accelerator on the path to the desired goal. Inspiration is a special moment in a rational process. The two are inseparable and complementary."[5]

Nothing in the present model prejudges the balance of roles, once having accepted the fact that we will make use of the computer in some way. There is still great freedom in deciding which aspects of each of the procedures we have identified should be machine-computed, and which require heavy or complete use of human abilities. Roughly speaking, Prediction will make maximum use of the computer, and Choice the least, with Search and Evaluation somewhere between. But the real resolution of this issue will be a personal decision in which the designer always retains the prerogative of using machine-aided procedures or his own intuitive capabilities at each step of the Problem-Solving Process.[6]

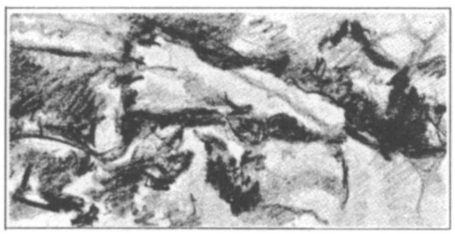
Some procedures will be general and applicable to a variety of design and planning problems; others will be more specific and applicable only to a special problem. Prediction procedures will be fairly specific in their application — for example, the prediction of structural behavior of a bridge, or of a sequence of spatial perceptions while driving along a highway. For Search and Evaluation procedures, generality is a matter of degree.

Highway location study: The problem was to locate a twenty-mile stretch of highway in Massachusetts beginning at Springfield and ending near Northampton.

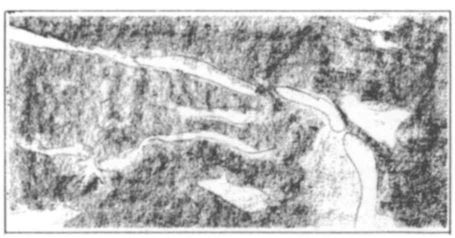
Twenty-six forces or requirements were determined for the location of the highway. Each of these requirements demanded a certain kind of route location (requirement No. 1, for example, "earthwork costs" or *the need to reduce earthwork costs* seeks a highway location through areas where land is flat).

1. **Earthwork Costs.**
2. **Comfort and Safety.**
3. **Regional Development.**
4. **Local Land Development.**
5. **Obsolescence.**
6. **Interference During Construction.**
7. **User Costs.**
8. **Services.**
9. **Travel Time.**
10. **Pavement and Subgrade Costs.**
11. **Drainage Patterns.**
12. **Bridge Costs.**
13. **Land Costs.**
14. **Eyesores.**
15. **Noise.**
16. **Air Pollution.**
17. **Weather Effects.**
18. **Non-Recompensable Public and Private Losses.**
19. **Public Financial Losses.**
20. **Major Current Traffic Desires.**
21. **Catchment Areas.**
22. **Local Accessibility and Integrity.**
23. **Future Transportation Systems.**
24. **Existing Transportation Systems.**
25. **Duplication of Facilities.**
26. **Self-Induced Congestion.**

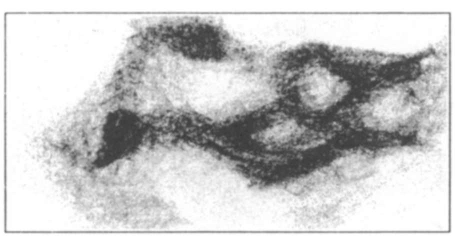
Twenty-six requirements represented as patterns in shades of gray over the terrain corresponding to the Springfield/Northampton area. Each diagram is a pattern of grays whose density varies over the complete range from white to black. This pattern is keyed to the base map of the terrain in such a way that a point marked black in the diagram for a particular requirement is a very good point for a highway location to pass through (with regard to that requirement).
Any point marked white is very bad as far as that requirement is concerned.



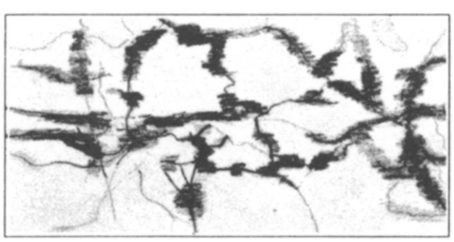
I. EARTHWORK COSTS



2. COMFORT AND SAFETY



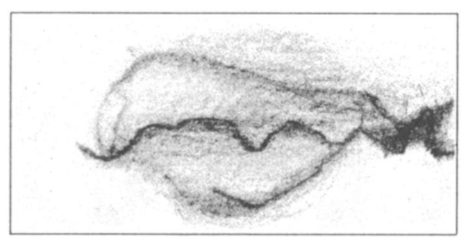
3. REGIONAL DEVELOPMENT



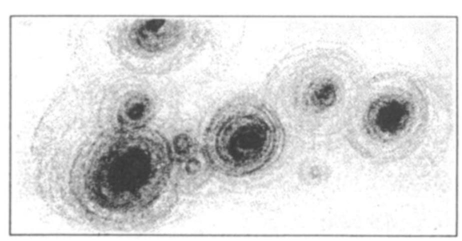
4. LOCAL LAND DEVELOPMENT

5. OBSOLESENCE

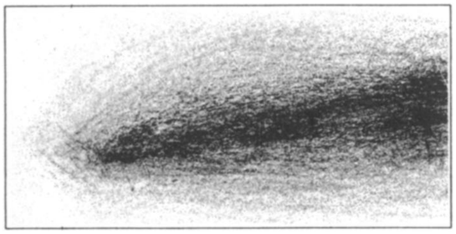
6. INTERFERENCE DURING CONSTRUCTION



7. USER COSTS



8. SERVICES



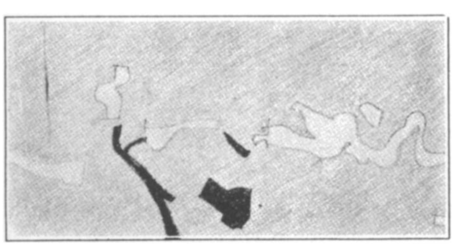
9. TRAVEL TIME

10. PAVEMENT AND SUBGRADE COSTS

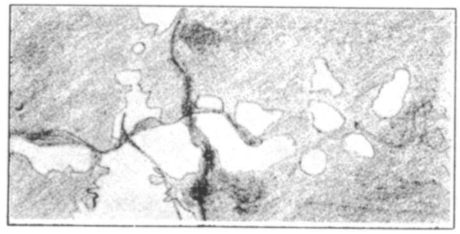
11. DRAINAGE PATTERNS

12. BRIDGE COSTS

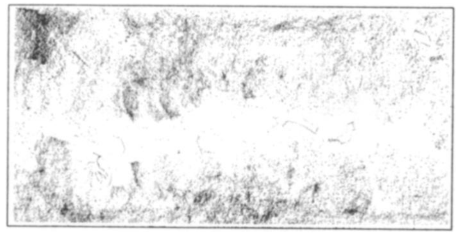
13. LAND COSTS



14. EYESORES



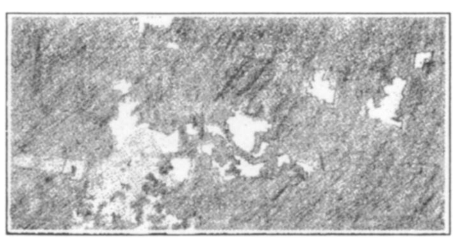
15. NOISE



16. AIR POLLUTION

17. WEATHER EFFECTS

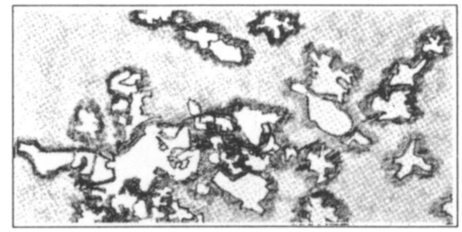
18. NON-RECOMPENSABLE PUBLIC AND PRIVATE LOSSES



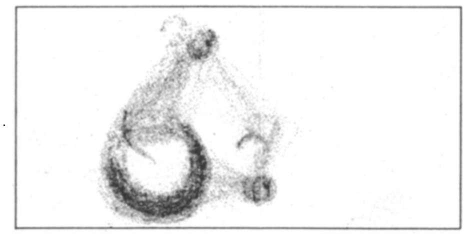
19. PUBLIC FINANCIAL LOSSES

20. MAJOR CURRENT TRAFFIC DESIRES

21. CATCHMENT AREAS



22. LOCAL ACCESSIBILITY AND INTEGRITY

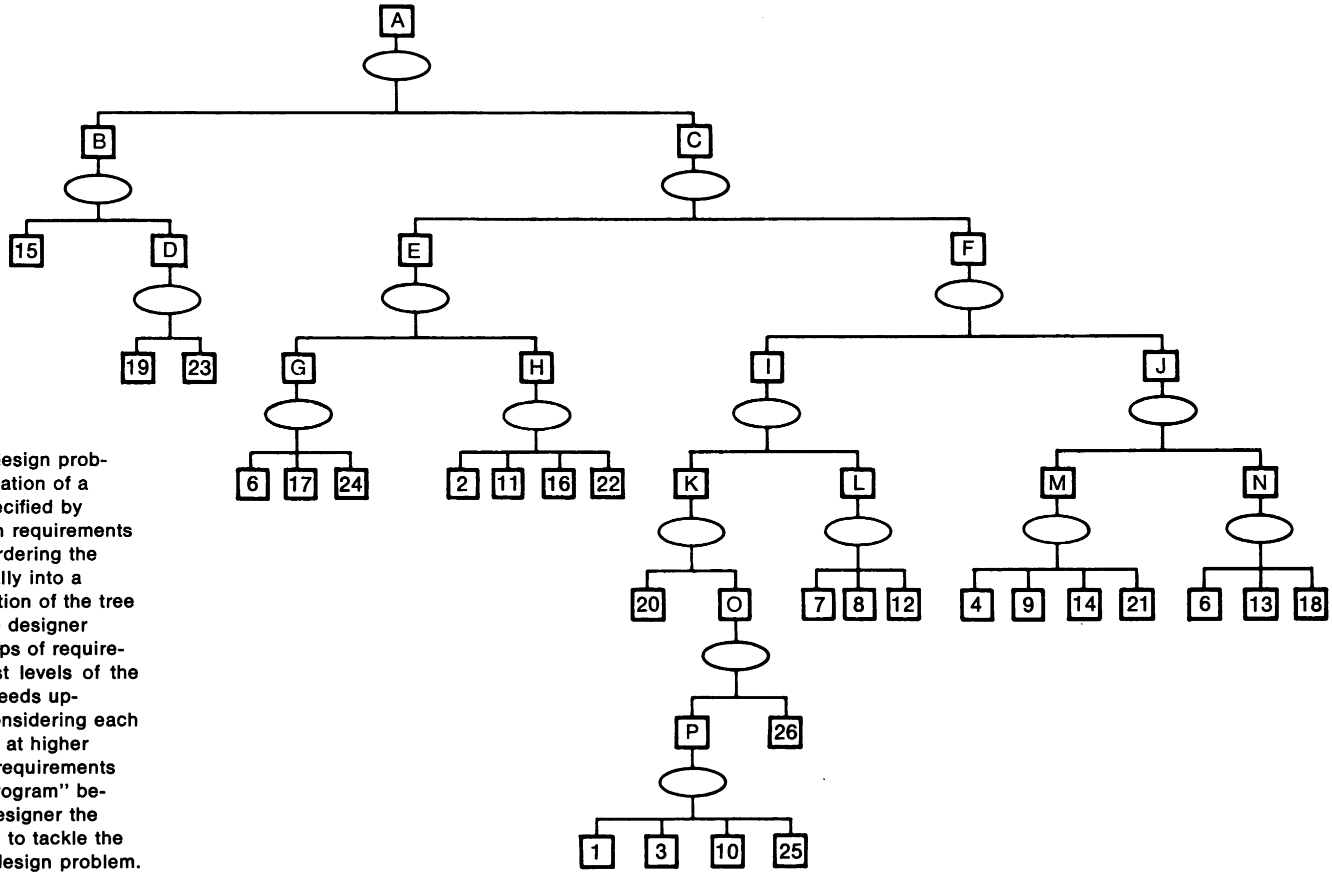


23. FUTURE TRANSPORTATION SYSTEMS

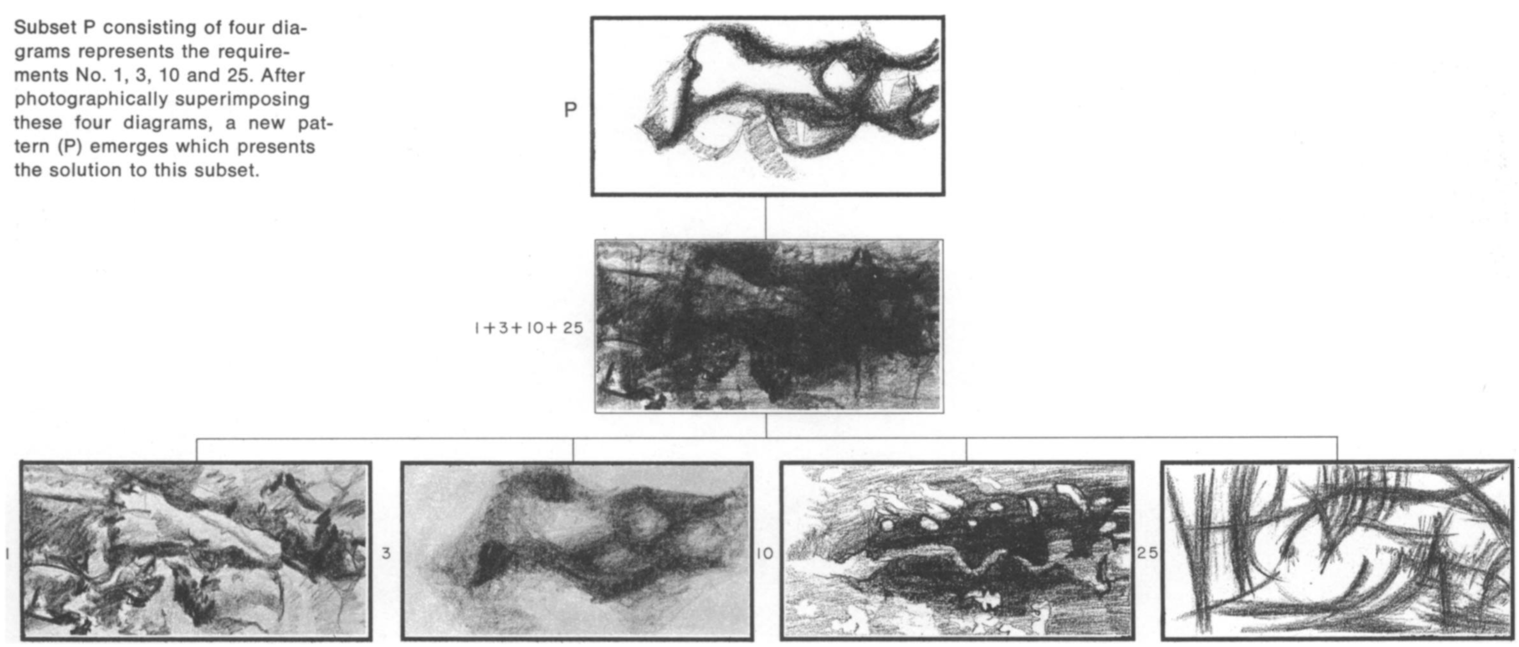
24. EXISTING TRANSPORTATION SYSTEMS

25. DUPLICATION OF FACILITIES

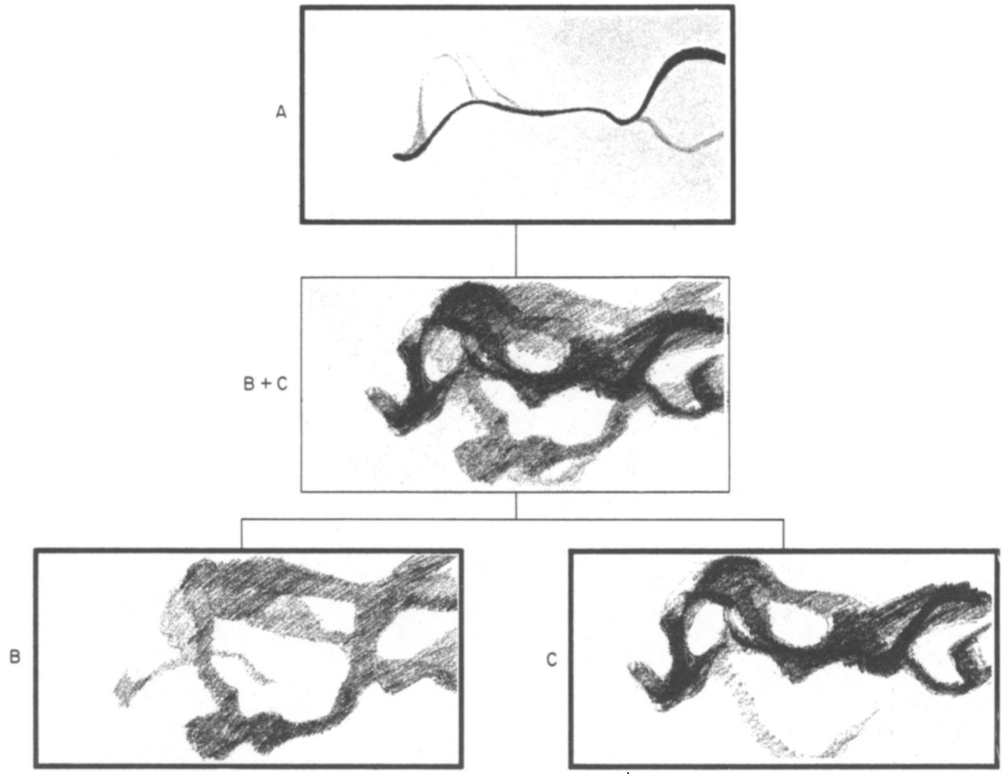
26. SELF-INDUCED CONGESTION



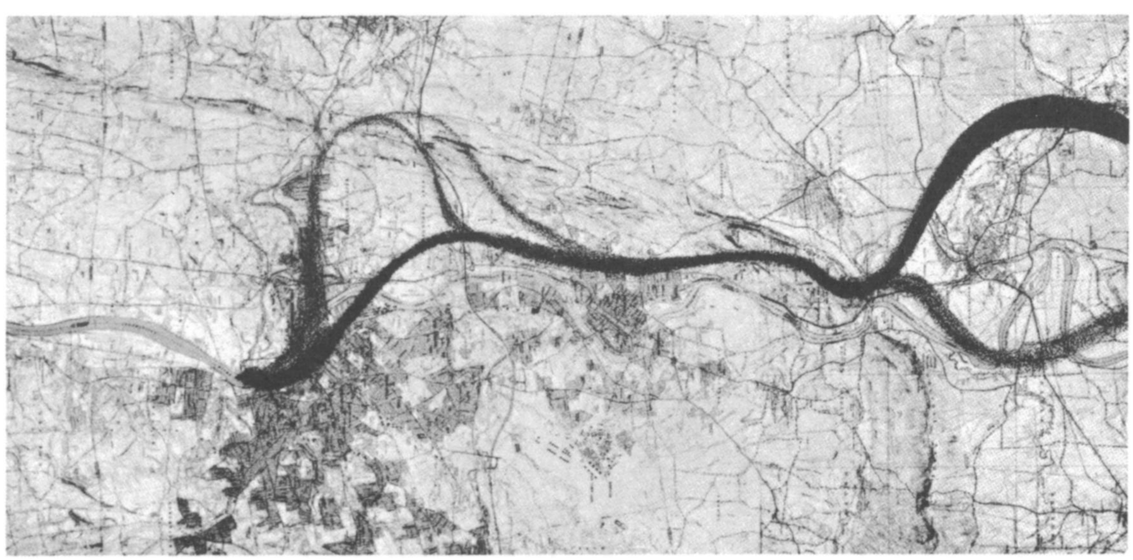
The structure of a design problem such as the location of a highway can be specified by grouping the design requirements into subsets, and ordering the subsets hierarchically into a "tree." The implication of the tree structure is that the designer starts with the groups of requirements at the lowest levels of the tree, and then proceeds upwards, gradually considering each of the other groups at higher levels. The tree of requirements can be called a "program" because it shows a designer the best order in which to tackle the requirements in a design problem.



Subset P consisting of four diagrams represents the requirements No. 1, 3, 10 and 25. After photographically superimposing these four diagrams, a new pattern (P) emerges which presents the solution to this subset.



All subsets in the tree of requirements are solved by superimposition until finally, after a series of fusions, one diagram remains showing only a pair of lines defining the best location for the highway.



The path shown here is the solution to the highway location problem.

There will clearly be some aspects of Search and Evaluation which are specific to a particular problem context; but there are also a number of "service" tools or procedures which can apply to a variety of contexts. For example: hill-climbing and gradient-seeking methods, mathematical programming, and Christopher Alexander's method of hierarchical decomposition are used as Search procedures.[7] In the area of Evaluation, scale construction procedures and many economic formulas will be generally applicable. Procedures for Choice, Metaprocedures, Goal Revision and Information Analysis will all have significant generality and will be transferable from one problem to another. These general procedures can be part of the "service" routines built into an integrated system.

A Problem-Solving Process must be flexible. The sequence of use of the procedures cannot be completely determined in advance, but must adjust as the designer's view of the problem evolves. The designer will continually revise his image of the problem. He will discover new objectives and revise old ones. He will discover that some actions are completely impossible, and he will suddenly invent actions which are radically different from those previously investigated. He also will occasionally discover that his whole approach needs to be discarded.

This model of a Problem-Solving Process explicitly provides for evolution of the designer's view of the problem. This is accomplished through goal revision, model revision, the continual development of new actions, and decomposition and restructuring procedures. Metaprocedures will be provided to determine the procedures to be used as the problem evolves. Particularly important is the ability to use alternative Search and Selection procedures at different levels of detail so that the designer can shift from gross concepts to detailed designing and back again.[8]

The designer cannot obtain an optimal action, only an optimal process. This model of a Problem-Solving Process recognizes explicitly that there is no single criterion for choosing the best action, that generation of actions is difficult, exploration of the full domain of possible actions is completely infeasible, and that determining the desirability of a single action is difficult. Thus, it accepts that the optimal plan or action can never be found. The model recognizes that it is the process of problem solving which is important: one can talk about the optimal allocation of problem-solving resources so that the best action is found within the constraints of the resources and capabilities of the Problem-Solving Process.

Procedures may be executed in sequence or in parallel. The availability of computers which can do many jobs simultaneously through multiprocessors, or through time-sharing, emphasizes the capability to apply procedures in parallel. Some procedures are clearly constrained to sequential relation; Selection can operate only after Search has produced some actions. But most of the procedures may operate in parallel — for example, while Search and Selection are executed, new information may be analyzed, triggering goal revision; at the same time, metaprocedures are operating to compute what should be done after completion of the current Search-Selection iteration.

This qualitative outline of the functions and characteristics necessary in a general Problem-Solving Process still requires a description of specific modules and routines to accomplish the general functions outlined here.

It is expected that this model of a Problem-Solving Process will be substantially revised as time goes on and more knowledge about the structure of a Problem-Solving Process is gained. More research is needed and the application of this Problem-Solving Process should lead to a better understanding of how complex problems can be approached.

REFERENCES

1. Miller, C. L., "An Integrated System for Civil Engineering Design," *Building Research* 3:2 (March-April 1966)

2. Minsky, M., "Steps toward artificial intelligence" in Edward Feigenbaum and Julian Feldman (eds.) *Computers and Thought*. New York: McGraw-Hill (1963), p. 409

3. Reitman, Walter R., *Cognition and Thought*. New York: Wiley (1965), p. 148

4. Minsky, M., op. cit., p. 408

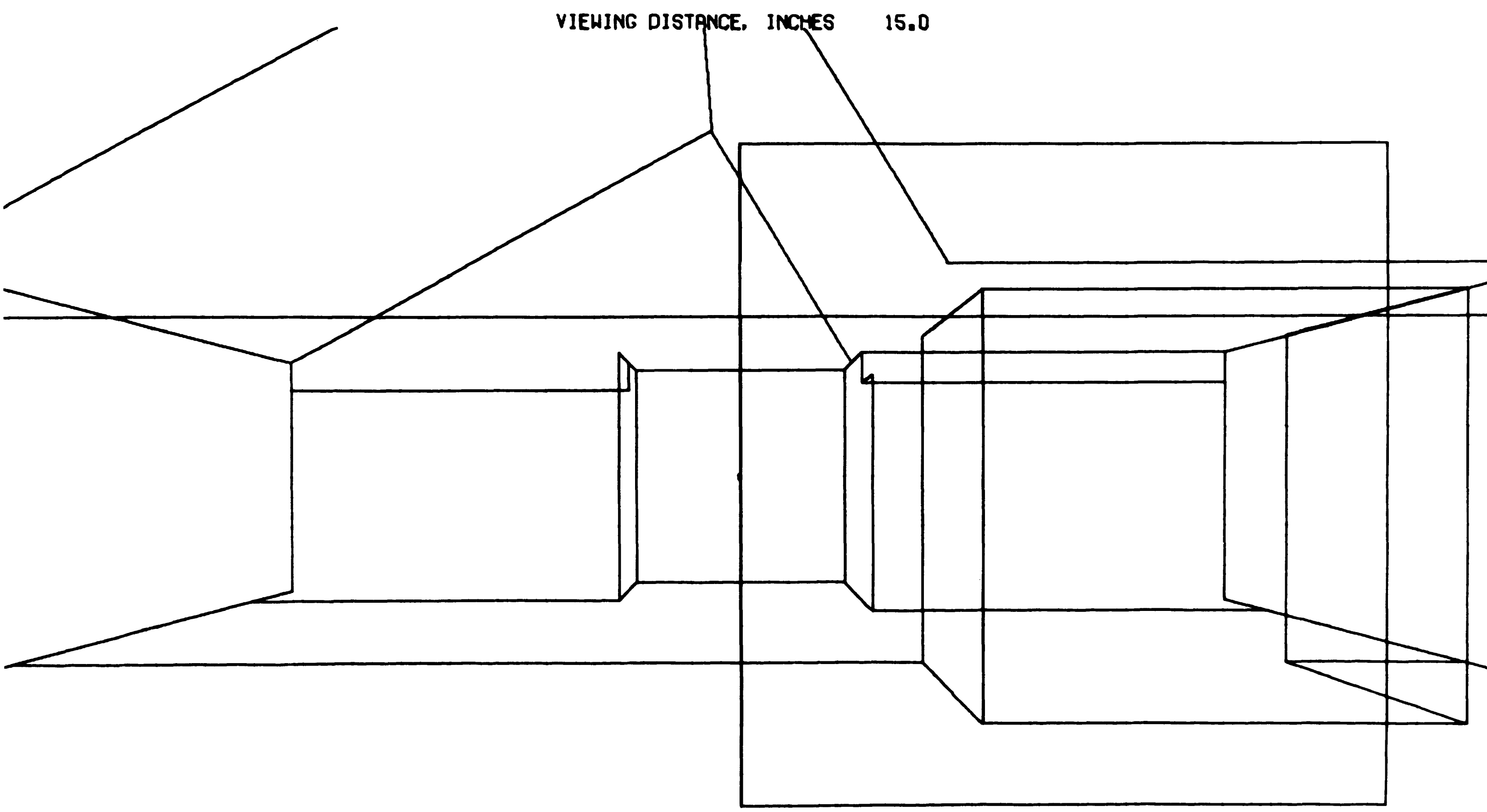
5. Chermayeff, S., *Architecture and the Computer*, the Proceedings of the First Boston Architectural Center Conference, Boston, Mass.: Boston Architectural Center (1964)

6. Manheim, M. L., "The role of the computer in the design process," *Building Research*, 1966, 3:2 (March-April)

7. Alexander, C., *Notes on the Synthesis of Form*. Cambridge, Mass.: Harvard University Press, 1964

8. Manheim, M. L., "Highway Route Location as a Hierarchically-Structured Sequential Decision Process." Research Report R64-15, Cambridge, Mass.: MIT Civil Engineering Systems Laboratory (1964)

One of a series of computer plots showing the inside of a small studio. Perspective computer programs driving the plotter allow the architect to pretest the design from all points of view and different distances.



COMPUTER-AUGMENTED DESIGN

A Case History in Architecture
by Allen Bernholtz and Edward Bierstone

Allen Bernholtz is assistant professor of Architecture and Computer Technology in the department of Architecture, Graduate School of Design, Harvard University. Formerly assistant professor of architecture at the University of Toronto, Canada, he obtained a grant from the National Research Council (Canada) for research on computers and architecture. Edward Bierstone, who collaborated with Mr. Bernholtz on this computer project, is a student of architecture at the University of Toronto.

In any design problem, certain requirements have to be met by the designer or architect. The interaction between the individual requirements makes it difficult to fulfill them all. When these requirements are few, as in the design of a simple product, the solution remains readily within the reach of the designer's immediate ability. But how does he proceed when confronted with a complex problem such as the design and location of highways, the architectural and organizational problems of hospitals or even the total environment for millions of people?

The complexity of such problems is so great that the designer and the architect will be unable to arrive at correct solutions unless a new way is found to structure the problem by breaking it down into smaller problems.

Christopher Alexander's method of the "hierarchical decomposition" and the computer program devised by him and Prof. Manheim break a design problem down into subsets. This method, here applied to the architectural project of a medium-size house, can also be used in the design of highways and, perhaps, even in the design of a whole city.

The complexity of today's design problems demands a revision of the processes of both analyzing the problem and synthesizing its solution. In the use of the computer as an aid to design lies the potential for such a revision and, therefore, the potential for the production of forms of consistent aesthetic and functional clarity.

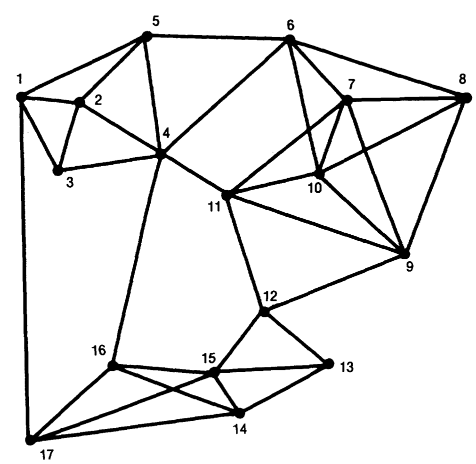
Using the IBM 7094 Computer at the University of Toronto's Institute of Computer Science, experiments have been undertaken with the application of the computer to the analysis of a design problem based on the inherent structure of the problem, and the formulation of a direction of attack for the synthesis of the form. An actual house design furnished a test case for the research and is used here as an example.[1]

COMPUTER-GENERATED FORMALIZATION OF THE DESIGN PROBLEM

Because of the great number of requirements to be simultaneously considered in an architectural design problem, the odds are heavily against the designer producing a solution which both satisfies the requirements, and especially their interactions, and at the same time achieves clarity of form. The designer usually attempts to structure the problem by verbally grouping these factors, or requirements, into "acoustics," "zoning," "circulation," and so on, and proceeds then to design by considering each *group* of requirements more or less independently.

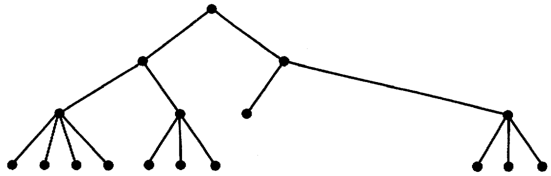
But, as Christopher Alexander has explained in his book, *Notes on the Synthesis of Form,*[2] this grouping of requirements by verbal concepts is irrelevant to the structure of the specific problem and introduces only further error into the already overly-complex design program. Alexander therefore proposed to describe the problem by a set of "misfit" factors, that is, by ways in which a form can "go wrong" and by a set of interactions between the misfits.

In the design problem discussed here, as in almost any design problem, it is not exactly the requirements or misfit factors that make up the project, although the designer has to find a form that finally reflects these requirements. After establishing all the requirements which have any influence or relation to the physical shape of the building, in the present case, 72 requirements, the problem is then to determine the links that connect them, to find the ways in which they interact. (Two misfits interact if the decisions the designer makes about the form to satisfy one, will affect the decisions he makes when considering the other.) This set of requirements and the set of links can be demonstrated by a linear graph where each requirement is represented by a point, and each interaction between requirements is represented with a link between the corresponding pair of points.

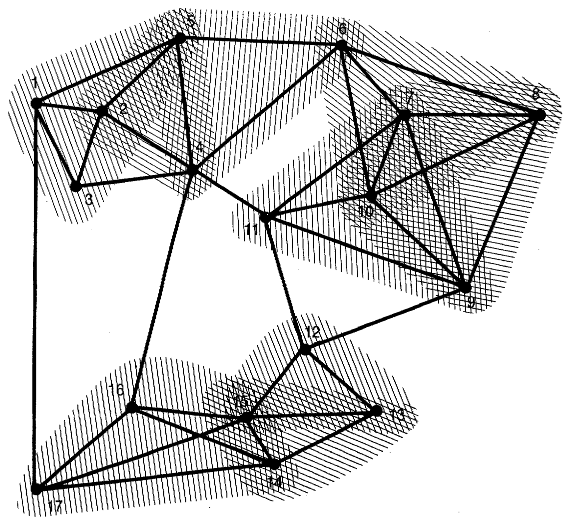


Linear graph for a design problem involving 17 requirements. Each numbered dot represents a need or requirement and each line or edge connecting these numbered dots indicates an interaction.

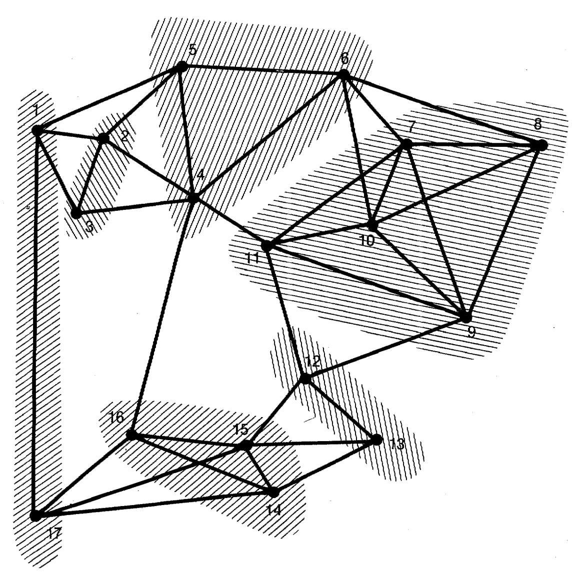
Such a graph presents us with a structural description of the functional total. This description then can be mathematically interpreted to suggest criteria for subdivision of the problem into subsystems, that is, functional units or components of the whole. Using a measure of independence to evaluate each possible way of subdividing the system, the designer then divides it into subsystems that are as independent as possible.



A computer program is used to produce a hierarchy by dividing the problem, in this example consisting of 17 requirements, into its two most independent parts, and continuing the process until a series of independent subsystems are achieved.



Graph showing the 17 requirements decomposed by a specific computer program into subsystems. Each subsystem consists of a minimum of three requirements, with each requirement linked to every other one within the subsystem.



A specific computer program is used to yield disjoint subsets of a problem consisting of 17 requirements. In contrast to the example above, this program does not produce subsets which overlap.

HIERARCHICAL DECOMPOSITION

Together with Marvin L. Manheim, Christopher Alexander developed in HIDECS 2 a computer program which divides such a system into its two most independent parts, and then repeats the process, operating on each of the subsets resulting from the previous operations, until complete subsystems (subsets with every pair of points linked) are produced.[3] This computer program yields a tree of hierarchy as the design program which we can solve by first finding solutions to the subproblems at the lowest level of the tree, and then proceeding to combine solutions, finally, into the complete form, in the order defined by the hierarchy. This program, however, has two faults: first, the decomposition of the system in binary steps results in misfit factors being considered only in the context of an immediately preceding subsystem but not in the context of the system as a whole; and second, the subsets of elements generated have no elements in common, although the most natural subsystems of a system may contain common elements.

To correct these weaknesses, Alexander developed four new programs under the title HIDECS 3. The programs, called BLDUP, STABL, SIMPX, and EQCLA, each decompose the set of interactions between the requirements in a single step, i.e. not into a hierarchy of subsystems.[4] In addition, two of them, SIMPX and EQCLA, generate subsystems which may overlap. (The names of these four computer programs conform to the custom in computer jargon of identifying these specific programs by relating the name of the program to the subject of the program – in this case EQCLA is a convenient name because it determines EQuivalent CLAsses of triangles.)

These programs, however, introduced a new problem: decomposition in a single step eliminates hierarchical ordering which is necessary in the design process because it allows for a design program starting with finding solutions for subsets beginning on the lowest level.

HIDECS 2, then, by decomposing the system in a series of binary steps yielding a hierarchy, sacrifices the unity of the system, while HIDECS 3, by decomposition in a single step, preserves the unity at the expense of the hierarchy. It became apparent that it was necessary to begin with a one-step decomposition, and then employ a procedure to recombine the resulting subsystems into a hierarchy.

HIERARCHICAL RECOMPOSITION

SIMPX, one of the HIDECS 3 computer programs, which generates complete subgraphs of a linear graph as subsystems, was used to yield the initial decomposition for this particular test problem. Each element in the subset is joined to every other element in that subset.

Since many of the subsystems generated by SIMPX contain common requirements, and since the aim of the computer decomposition is the production of subsets which the designer may solve independently, a natural criterion for the recomposition of these subsystems into a "tree" capable of being used as a design program is the combination of subsystems containing common requirements. The HIDECS 3 program BLDUP can be used for this purpose because it divides the elements into disjoint subsets (having no elements in common), so that in a decomposition the subsets do not overlap.

The subsystems yielded by the SIMPX program, that is the decomposition method which yields overlapping subsystems, are numbered and considered as single elements; two such elements are considered linked if they contain one or more common misfit factors. This information, then, defines a linear graph and its associated matrix of linked elements. BLDUP, the method of decomposition which yields disjointed subsets, is then used to group the initial subsystems obtained by SIMPX. The process is performed twice more, the first time considering two elements as linked if they contain two or more misfit factors, and the second time considering two elements as linked if they contain three or more common misfit factors.* Of the three outputs, the one which yields the best composition of subsystems, from the point of view of the number of subproblems that may be conveniently combined in a single step, is used for the second lowest level of the hierarchy.

Now, considering the disjointed subsystems as elements, and defining links as above, BLDUP is again run three times, and one of the outputs is chosen for the third level of the hierarchy. The process is continually repeated until the program yields a number of subsystems sufficiently small to be combined in a single step, into the entire system. A hierarchy defining the order of recomposition of the subproblems is thus produced.

THE DESIGN OF A HOUSE AS A SAMPLE PROBLEM

The following is a list of misfit factors which are related to or influence the architectural form of this particular house.

Since Fitness of Form, "the relation of mutual adaptability of form and requirement," is better noticed or described when it is negative or absent, the requirements for this design problem are here introduced as negative factors.

*The number of common elements in a subsystem on a particular level determines the numbers likely to be used to define interactions between subsystems. The definition of interactions for a level containing subsystems with a great many common elements may be based on a number of common elements greater than three.

This in turn means that good adaptability of form to requirement is achieved by neutralizing the discrepancies or forces which cause misfit in the design process.

MISFIT FACTORS

1. Inadequate protection for cars.
2. Large area to be snow-shovelled in winter.
3. No provision for hobbies.
4. Siting of dwelling hinders development of vegetable, flower gardens.
5. Eastern Canadian woods little used in design and furnishing.
6. Habitable spaces not naturally well-ventilated.
7. No provision for reception of delivery, mail, parcels.
8. No exterior space for rest.
9. No exterior conversation space.
10. No exterior space for children's play.
11. No protection during arrival and entry.
12. No meeting space for arrivals.
13. No provision for storage of outdoor clothes.
14. No protection from dust and dirt.
15. No protection from animals, insects, vermin.
16. No protection from human intruders.
17. No separation of children from vehicles.
18. Poor fire protection.
19. Transmission of exterior noises.
20. Transmission of interior noises to quiet areas.
21. Difficult emergency access.
22. Difficult emergency escape.
23. Access from car to dwelling involves long distance.
24. Inconvenient system of garbage disposal.
25. Inadequate provision for garbage collection.
26. Inadequate food storage.
27. Food processing inconvenient.
28. Food conservation inadequately provided for.
29. Inadequate storage for culinary and eating equipment.
30. Poor provision for cleaning of culinary and eating equipment.
31. Inadequate eating facilities.
32. Poor accessibility to public utilities.
33. Poor sun control.
34. Dwelling enclosure vulnerable to rain.
35. Poor wind control.
36. Little resistance to changes in temperature, humidity.
37. Inaccessibility to top of dwelling.
38. Heating difficult to control and maintain.
39. Inadequate water-heating system.

40. Inconvenience in reaching telephone.
41. Poor provision for radio, recordings.
42. Poor provision for television.
43. Inadequate facilities for washing of laundry.
44. Inadequate facilities for drying of laundry.
45. Inadequate facilities for ironing of laundry.
46. Inadequate or inconvenient facilities for washing.
47. Inadequate or inconvenient facilities for bathing and showering.
48. Inadequate or inconvenient facilities for elimination of excreta.
49. No provision for storage of medicinal supplies.
50. No provision for storage of toiletries.
51. Inadequate facilities for personal care.
52. Poor facilities for solitude for members of family.
53. Poor facilities for rest, sleep for members of family.
54. Poor love-making facilities for parents.
55. Inadequate provision for storage of personal possessions.
56. Poor facilities for dressing and undressing.
57. Poor provision for storage of clean clothing.
58. Poor provision for storage of clothing in use.
59. Poor provision for storage of clothing to be washed.
60. Poor provision for storage of linen.
61. Inadequate facilities for studying and writing.
62. Inadequate facilities for reading.
63. Poor provision for storage of gardening and exterior maintenance equipment.
64. No separation of pets from vehicles.
65. No provision for storage of domestic cleaning equipment.
66. No provision for storage of domestic repair equipment.
67. No provision for storage of sports equipment.
68. No provision for general storage, luggage, etc.
69. No provision for storage of clothing in seasonal use.
70. Inadequate facilities for family activities.
71. Inadequate facilities for activity with guests.
72. Inadequate facilities for repair of clothing.

SET OF LINKS BETWEEN MISFIT FACTORS

Each of the 72 misfit factors connects or interacts with one or more of the other factors. It is essential to find and identify those links which most definitely relate to the structure of the problem. The following links have been determined and used for this test problem:

Misfit factor 1, *Inadequate protection for cars,* is connected to:

2 (Large area to be snow-shovelled in winter)
11 (No protection during arrival and entry)
17 (No separation of children from vehicles)
19 (Transmission of exterior noises)
23 (Access from car to dwelling involves long distance)
63 (Poor provision for storage of gardening and exterior maintenance equipment)
64 (No separation of pets from vehicles)
67 (No provision for storage of sports equipment)

Misfit factor 2, *Large area to be snow-shovelled in winter,* is connected to:

1 (Inadequate protection for cars)
7 (No provision for reception of delivery, mail, parcels)
11 (No protection during arrival and entry)
21 (Difficult emergency access)
23 (Access from car to dwelling involves long distance)
25 (Inadequate provision for garbage collection)
37 (Inaccessibility to top of dwelling)
63 (Poor provision for storage of gardening and exterior maintenance equipment)

Misfit factor 3, *No provision for hobbies,* is connected to:

6 (Habitable spaces not naturally well-ventilated)
7 (No provision for reception of delivery, mail, parcels)
8 (No exterior space for rest)
14 (No protection from dust and dirt)
18 (Poor fire protection)
20 (Transmission of interior noises to quiet areas)
24 (Inconvenient system of garbage disposal)
40 (Inconvenience in reaching telephone)
44 (Inadequate facilities for drying of laundry)
46 (Inadequate or inconvenient facilities for washing)
53 (Poor facilities for rest, sleep for members of family)
61 (Inadequate facilities for studying and writing)
62 (Inadequate facilities for reading)
66 (No provision for storage of domestic repair equipment)

Misfit factor 4, *Siting of dwelling hinders development of vegetable, flower gardens,* is connected to:

8 (No exterior space for rest)
9 (No exterior conversation space)
10 (No exterior space for children's play)
15 (No protection from animals, insects, vermin)
33 (Poor sun control)
35 (Poor wind control)
40 (Inconvenience in reaching telephone)
46 (Inadequate or inconvenient facilities for washing)
63 (Poor provision for storage of gardening and exterior maintenance equipment)

Misfit factor 5, *Eastern Canadian woods little used in design and furnishing,* is connected to:
6 (Habitable spaces not naturally well-ventilated)
19 (Transmission of exterior noises)
20 (Transmission of interior noises to quiet areas)
33 (Sun uncontrolled)
34 (Dwelling enclosure vulnerable to rain)
35 (Wind cannot be controlled)
36 (Little resistance to changes in temperature, humidity)
38 (Heating difficult to control and maintain)

Misfit factor 6, *Habitable spaces not naturally well-ventilated,* is connected to:
3 (No provision for hobbies)
5 (Eastern Canadian woods little used in design and furnishing)
14 (No protection from dust and dirt)
15 (No protection from animals, insects, vermin)
16 (No protection from human intruders)
19 (Transmission of exterior noises)
20 (Transmission of interior noises to quiet areas)
24 (Inconvenient system of garbage disposal)
27 (Food processing inconvenient)
33 (Sun uncontrolled)
34 (Dwelling enclosure vulnerable to rain)
35 (Wind cannot be controlled)
36 (Little resistance to changes in temperature, humidity)
38 (Heating difficult to control and maintain)
44 (Inadequate facilities for drying of laundry)
48 (Inadequate or inconvenient facilities for elimination of excreta)

Misfit factor 7, *No provision for reception of delivery, mail, parcels,* is connected to:
2 (Large area to be snow-shovelled in winter)
3 (No provision for hobbies)
11 (No protection during arrival and entry)
12 (No meeting space for arrivals)
14 (No protection from dust and dirt)
15 (No protection from animals, insects, vermin)
16 (No protection from human intruders)
17 (No separation of children from vehicles)
19 (Transmission of exterior noises)
26 (Inadequate food storage)
32 (Poor accessibility to public utilities)
34 (Dwelling enclosure vulnerable to rain)
35 (Wind cannot be controlled)
64 (No separation of pets from vehicles)
68 (No provision for general storage, luggage, etc.)

The rest of the interactions between the misfit factors are listed numerically only:

Misfit factor 8 is connected to:
3, 4, 9, 10, 13, 14, 15, 16, 19, 52, 53, 54, 61, 62.

Misfit factor 9 is connected to:
4, 8, 10, 13, 14, 15, 16, 19, 40, 53, 61, 62, 70, 71.

Misfit factor 10 is connected to:
4, 8, 9, 13, 14, 15, 16, 17, 19, 27, 46, 53, 61, 62, 67.

Misfit factor 11 is connected to:
1, 2, 7, 12, 14, 15, 19, 21, 22, 34, 35.

Misfit factor 12 is connected to:
7, 11, 13, 14, 15, 16, 19, 46, 48.

Misfit factor 13 is connected to:
8, 9, 10, 12, 46, 71.

Misfit factor 14 is connected to:
3, 6, 7, 8, 9, 10, 11, 12, 15, 16, 19, 24, 25, 34, 35, 44.

Misfit factor 15 is connected to:
4, 6, 7, 8, 9, 10, 11, 12, 14, 16, 19, 24, 25, 26, 28, 34, 35, 44, 57, 60, 69.

Misfit factor 16 is connected to:
6, 7, 8, 9, 10, 12, 14, 15, 19, 21, 34, 35.

Misfit factor 17 is connected to:
1, 7, 10, 23, 64.

Misfit factor 18 is connected to:
3, 20, 36, 38, 39.

Misfit factor 19 is connected to:
1, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 23, 34, 35, 36.

Misfit factor 20 is connected to:
3, 5, 6, 18, 38, 40, 41, 42, 72.

Misfit factor 21 is connected to:
2, 11, 16, 22.

Misfit factor 22 is connected to:
11, 21.

Misfit factor 23 is connected to:
1, 2, 17, 19, 64.

Misfit factor 24 is connected to:
3, 6, 14, 15, 25, 27, 30, 46, 63.

Misfit factor 25 is connected to:
2, 14, 15, 24.

Misfit factor 26 is connected to:
7, 15, 27, 28, 29, 31.

Misfit factor 27 is connected to:
6, 10, 24, 26, 28, 29, 30, 31, 40, 46, 49, 65.

Misfit factor 28 is connected to:
15, 26, 27, 29, 31.

Misfit factor 29 is connected to:
26, 27, 28, 30, 31.

Misfit factor 30 is connected to:
24, 27, 29, 31, 40, 46, 65.

Misfit factor 31 is connected to:
26, 27, 28, 29, 30, 40, 46, 70, 71.

Misfit factor 32 is connected to:
7, 38, 63.

Misfit factor 33 is connected to:
4, 5, 6, 36.

Misfit factor 34 is connected to:
5, 6, 7, 11, 14, 15, 16, 19, 35, 36.

Misfit factor 35 is connected to:
4, 5, 6, 7, 11, 14, 15, 16, 19, 34, 36.

Misfit factor 36 is connected to:
5, 6, 18, 19, 33, 34, 35, 38.

Misfit factor 37 is connected to:
2, 63.

Misfit factor 38 is connected to:
5, 6, 18, 20, 32, 36, 39.

Misfit factor 39 is connected to:
18, 38, 43, 46, 47.

Misfit factor 40 is connected to:
3, 4, 9, 20, 27, 30, 31, 41, 42, 43, 45, 61, 62, 70, 71.

Misfit factor 41 is connected to:
20, 40, 42, 70, 71.

Misfit factor 42 is connected to:
20, 40, 41, 70, 71.

Misfit factor 43 is connected to:
39, 40, 44, 46, 59.

Misfit factor 44 is connected to:
3, 6, 14, 15, 43, 45, 57.

Misfit factor 45 is connected to:
40, 44, 46, 57, 60, 69, 72.

Misfit factor 46 is connected to:
3, 4, 10, 12, 13, 24, 27, 30, 31, 39, 43, 45, 47, 48, 49, 50, 51, 52, 53, 54, 56, 60, 63.

Misfit factor 47 is connected to:
39, 46, 48, 50, 51, 53, 54, 56, 57, 58, 60.

Misfit factor 48 is connected to:
6, 46, 47, 50, 52, 53, 54, 70, 71.

Misfit factor 49 is connected to:
27, 46, 50, 51.

Misfit factor 50 is connected to:
46, 47, 48, 49, 51, 56.

Misfit factor 51 is connected to:
46, 47, 49, 50, 55, 56.

Misfit factor 52 is connected to:
8, 46, 48, 53, 54, 61, 62, 70, 71.

Misfit factor 53 is connected to:
3, 8, 9, 10, 46, 47, 48, 52, 54, 56, 60, 70, 71.

Misfit factor 54 is connected to:
8, 46, 47, 48, 52, 53, 56, 70, 71.

Misfit factor 55 is connected to:
51, 56.

Misfit factor 56 is connected to:
46, 47, 50, 51, 53, 54, 55, 57, 58, 59, 68.

Misfit factor 57 is connected to:
15, 44, 45, 47, 56, 58, 59, 60, 69.

Misfit factor 58 is connected to:
47, 56, 57, 59.

Misfit factor 59 is connected to:
43, 56, 57, 58.

Misfit factor 60 is connected to:
15, 45, 46, 47, 53, 57.

Misfit factor 61 is connected to:
3, 8, 9, 10, 40, 52, 62, 70, 71.

Misfit factor 62 is connected to:
3, 8, 9, 10, 40, 52, 61, 70, 71.

Misfit factor 63 is connected to:
1, 2, 4, 24, 32, 37, 46, 66, 67.

Misfit factor 64 is connected to:
1, 7, 17, 23.

Misfit factor 65 is connected to:
27, 30, 66.

Misfit factor 66 is connected to:
3, 63, 65, 72.

Misfit factor 67 is connected to:
1, 10, 63, 68.

Misfit factor 68 is connected to:
7, 12, 56, 67, 69.

Misfit factor 69 is connected to:
15, 45, 57, 68, 72.

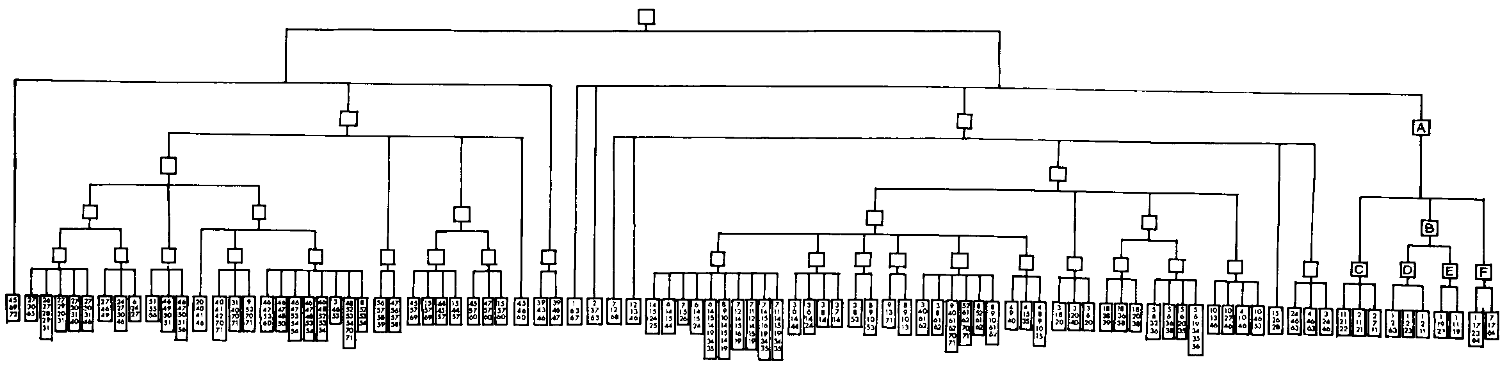
Misfit factor 70 is connected to:
9, 31, 40, 41, 42, 48, 52, 53, 54, 61, 62, 71.

Misfit factor 71 is connected to:
9, 13, 31, 40, 41, 42, 48, 52, 53, 54, 61, 62, 70.

Misfit factor 72 is connected to:
20, 45, 66, 69.

The above links were determined directly as the data cards were punched. Each link decision was made twice, and the computer was programmed to eliminate those links defined only once in the process, to produce a symmetric matrix. In spite of a rapid definition of this data, however, an examination of the final hierarchy produced testified to the logic of the list of requirements and their estimated interaction.

Data based on subsystems having two or more common elements were used for the lowest level of the hierarchy while data based on subsystems having three or more common elements were used for the second and third levels. Data based on subsystems having two or more common elements were used for the fourth level, and finally, data based on subsystems having one or more common elements were used for the fifth level, which is composed of two large subsystems of the system of misfits and links.



Level by level recomposition into a hierarchy of the subsets representing the architectural problem of a house design. This hierarchy constitutes the design program of the architect.

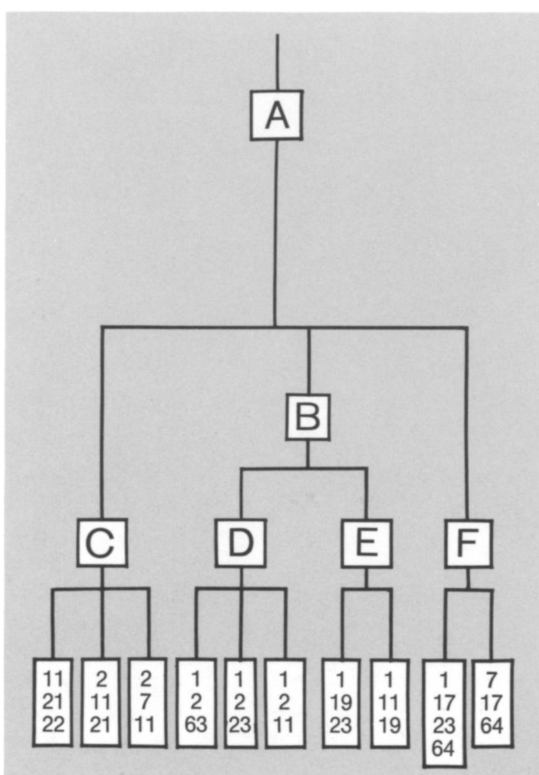


Diagram showing Subsystem A (including subsets B, C, D, E and F) of the above hierarchy. Subsystem A represents the garage and entrance area (Misfit factors 1, 2, 7, 11, 17, 19, 21, 22, 23, 63, 64).

HIERARCHICAL SYNTHESIS OF THE FORM

Because the subsystems in the lowest level of the hierarchy contained a convenient number of misfit factors (from 3 to 8), diagrams for the synthesis were commenced here.

The formal diagrams which satisfy the misfit factors and their links found in each subsystem in a level of the hierarchy were produced before an attempt was made to solve any subsystems in the next higher level. This method was used in order to avoid biasing subsystems by decisions made at a higher level. The synthetic process is thus continued until a diagram of a form satisfying the complete system of misfits and links is produced.

The synthetic process applied to the test problem, is illustrated in the following diagrams and explanations for that part of the hierarchy building up to subsystem A. It is significant that comprehensive analysis of a problem like the house, seemingly too simple to warrant this analytic process, reveals that even so-called trivial problems contain too many factors and interactions to be satisfactorily handled by the unassisted designer.

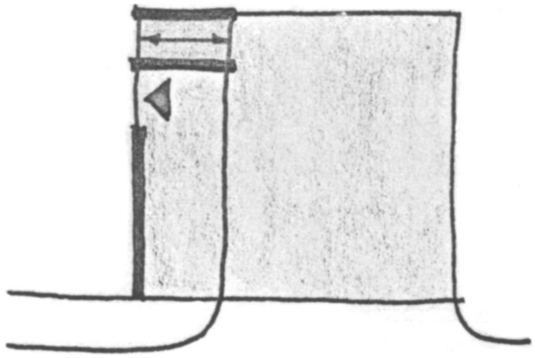
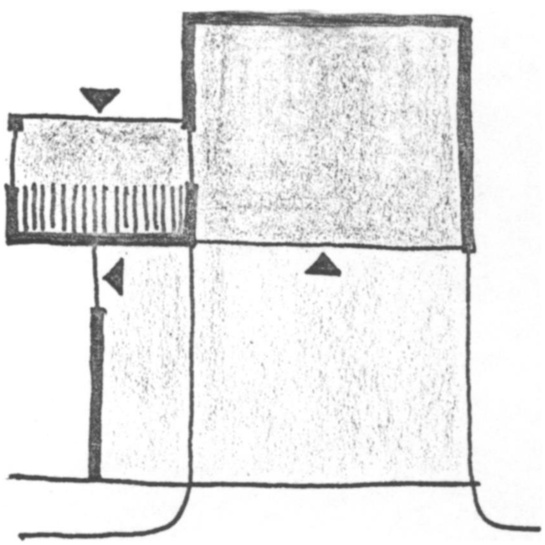


Diagram showing solution for Subsystem C. The architect begins by finding solutions for the subsets on the lowest level of the hierarchy by satisfying all the requirements in this subset (2, 7, 11, 21, 22).

SUBSYSTEM C — Misfit factors:

2. Large area to be snow-shovelled in winter.
7. No provision for reception of delivery, mail, parcels.
11. No protection during arrival and entry.
21. Difficult emergency access.
22. Difficult emergency escape.

Misfit factors 2 and 11 call for a roofed driveway, no longer than necessary for a car, and a separate roofed sidewalk for pedestrian arrival, walled on the side remote from the driveway. A wide sidewalk satisfied 21 and 22, providing ample space for emergency access and escape, even if the driveway is occupied. Misfit 7 suggests a delivery chute, opening to the driveway at its upper left corner for distinctness from the entry and convenience for the driver of the delivery vehicle.

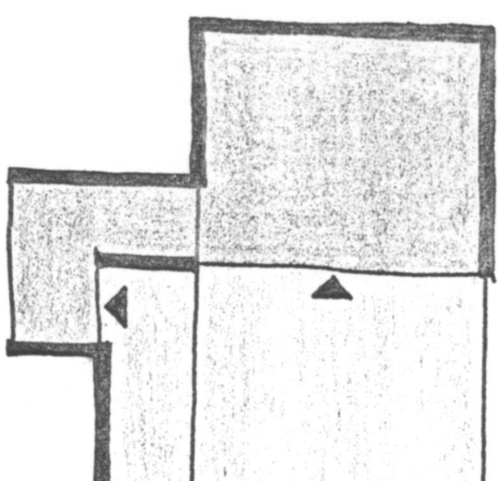


Solution for Subsystem D.

SUBSYSTEM D — Misfit factors:

1. Inadequate protection for cars.
2. Large area to be snow-shovelled in winter.
11. No protection during arrival and entry.
23. Access from car to dwelling involves long distance.
63. Poor provision for storage of gardening and exterior maintenance equipment.

Misfit factors 1, 2 and 11 call for a garage (preferably double), a roofed driveway slightly longer than a car, and a separate roofed sidewalk for pedestrian arrival, walled on the side remote from the driveway. Misfit factor 23 is best satisfied by entries to the dwelling on the driver's side, from both the driveway and the garage. A space for storage of gardening and exterior maintenance equipment, accessible from the dwelling's interior and opening on one side directly to the exterior and on the other through the entry from the garage solves Misfit factor 63, and provides a lock between the garage and dwelling to prevent transfer of fumes.



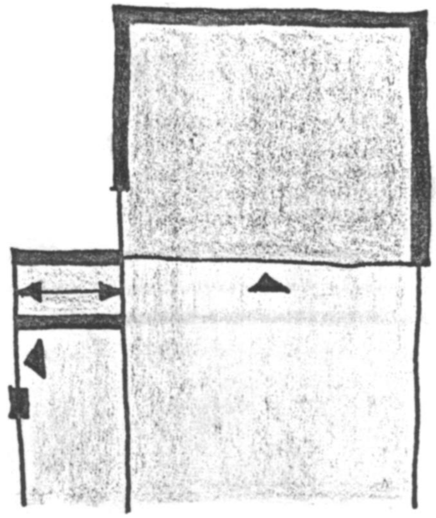
Solution for Subsystem E.

SUBSYSTEM E — Misfit factors:

1. Inadequate protection for cars.
11. No protection during arrival and entry.
19. Transmission of exterior noises.
23. Access from car to dwelling involves long distance.

Misfit factors 1, 11 and 23 call for a garage, roofed driveway, separate roofed sidewalk walled on the side remote from the driveway, and entries to the dwelling from the driver's side, from the driveway and the garage. Misfit 19 demands a buffer zone between the interior of the dwelling and the garage and exterior.

Solution for Subsystem F.



SUBSYSTEM F

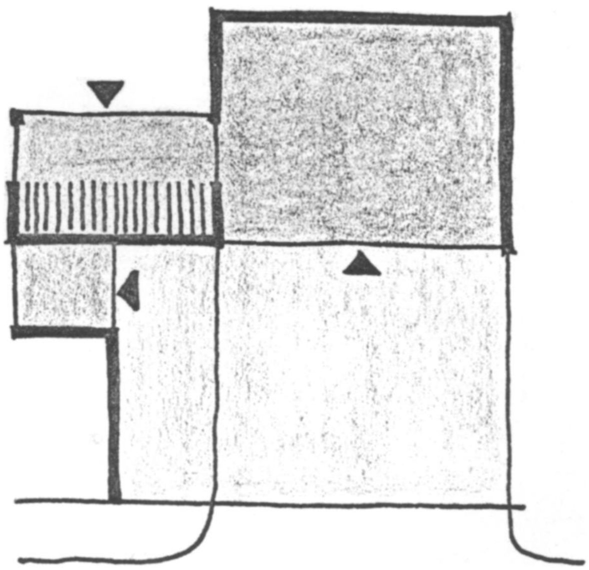
Misfit factors:

1. Inadequate protection for cars.
7. No provision for reception of delivery, mail, parcels.
17. No separation of children from vehicles.
23. Access from car to dwelling involves long distance.
64. No separation of pets from vehicles.

Misfit factors 1 and 23 call for a garage and entries to the dwelling from the driver's side, from both the driveway and garage. Misfit factor 7 is satisfied by a delivery chute opening at the upper left corner of the driveway, and misfit factors 17 and 64 by a play area separated from the driveway and road, and by a wide sidewalk to the dwelling entry.

SUBSYSTEM B *(Subsystems D plus E)*

The space for storage of gardening and maintenance equipment doubles as a noise buffer between the garage and dwelling, and a separate buffer is provided for the exterior entry.



Solution for Subsystem B.

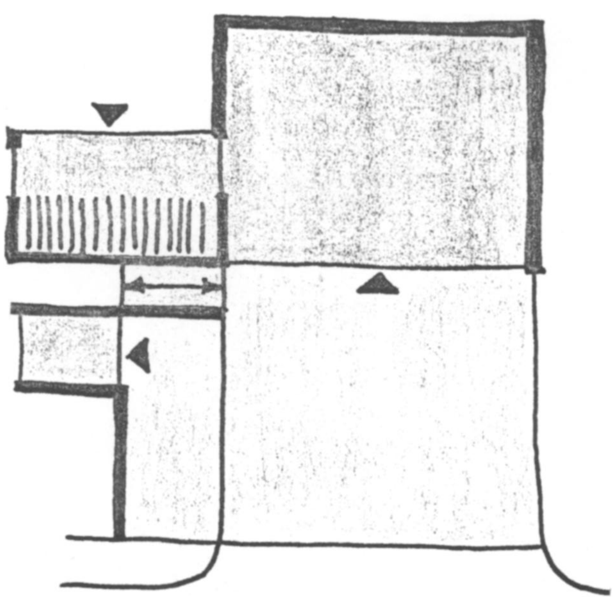
SUBSYSTEM A *(Subsystems B plus C plus F)*

The delivery chute and entry through the storage space from a service area are separated by an interior wall from the main entry to provide functional clarity.

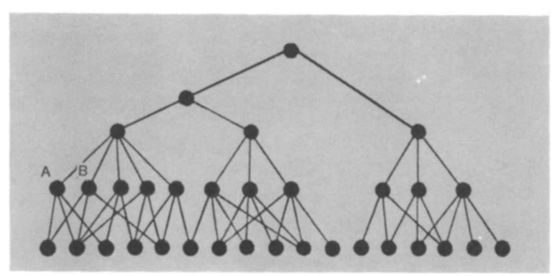
THE RECOMPOSITION IS A LATTICE

It should be noted that the diagram of the hierarchical design program is itself a linear graph, whose points are the subsystems and whose edges join these subsystems and indicate the order of recomposition. The graph of this hierarchy, furthermore, does not represent the complete hierarchy of the problem, as it lacks the bottom level, the individual misfit factors and the lines joining them into subsystems which were generated by the initial run of the SIMPX program.

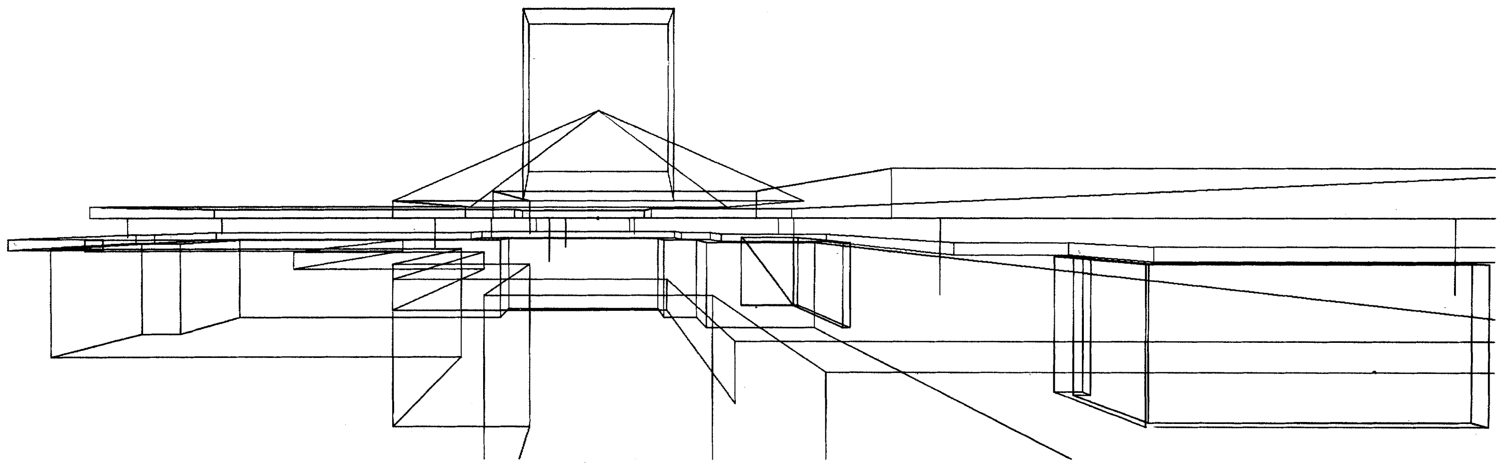
If, however, we include the bottom level (the individual misfits) of the hierarchy produced by the outlined method, the graph is no longer a pure tree, but assumes the form of the illustration below , as a result of common elements in some of the subsystems.

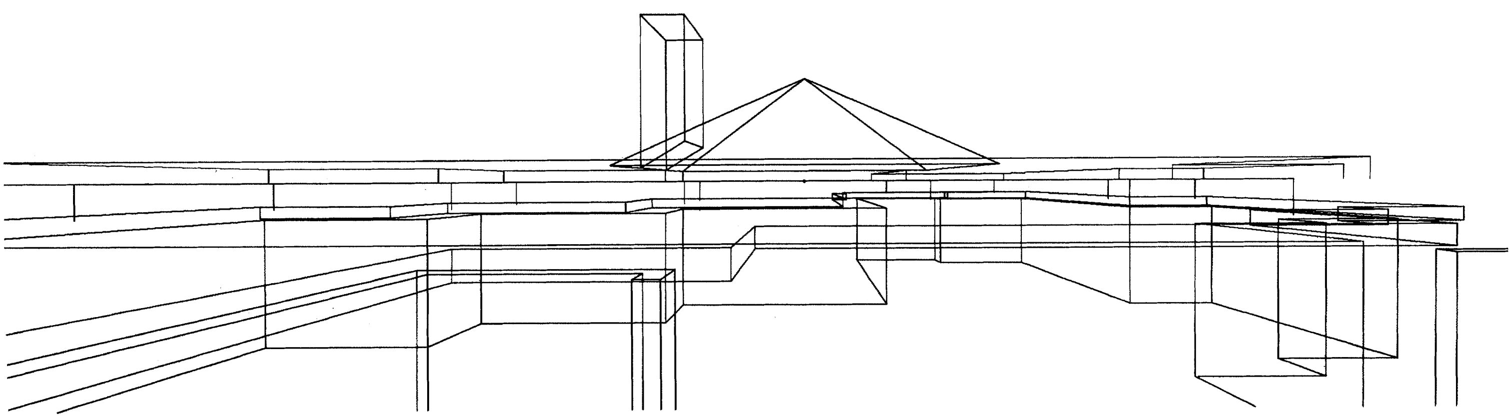


Solution for Subsystem A.



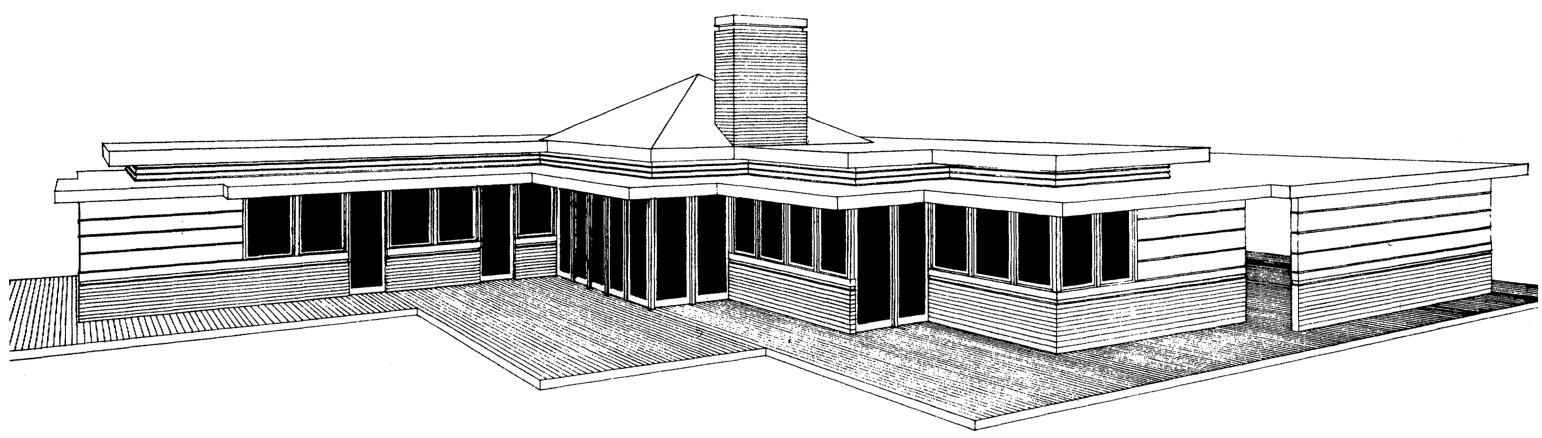
The design problem as a lattice. Requirements of Sub – systems A and B overlap in this diagram.

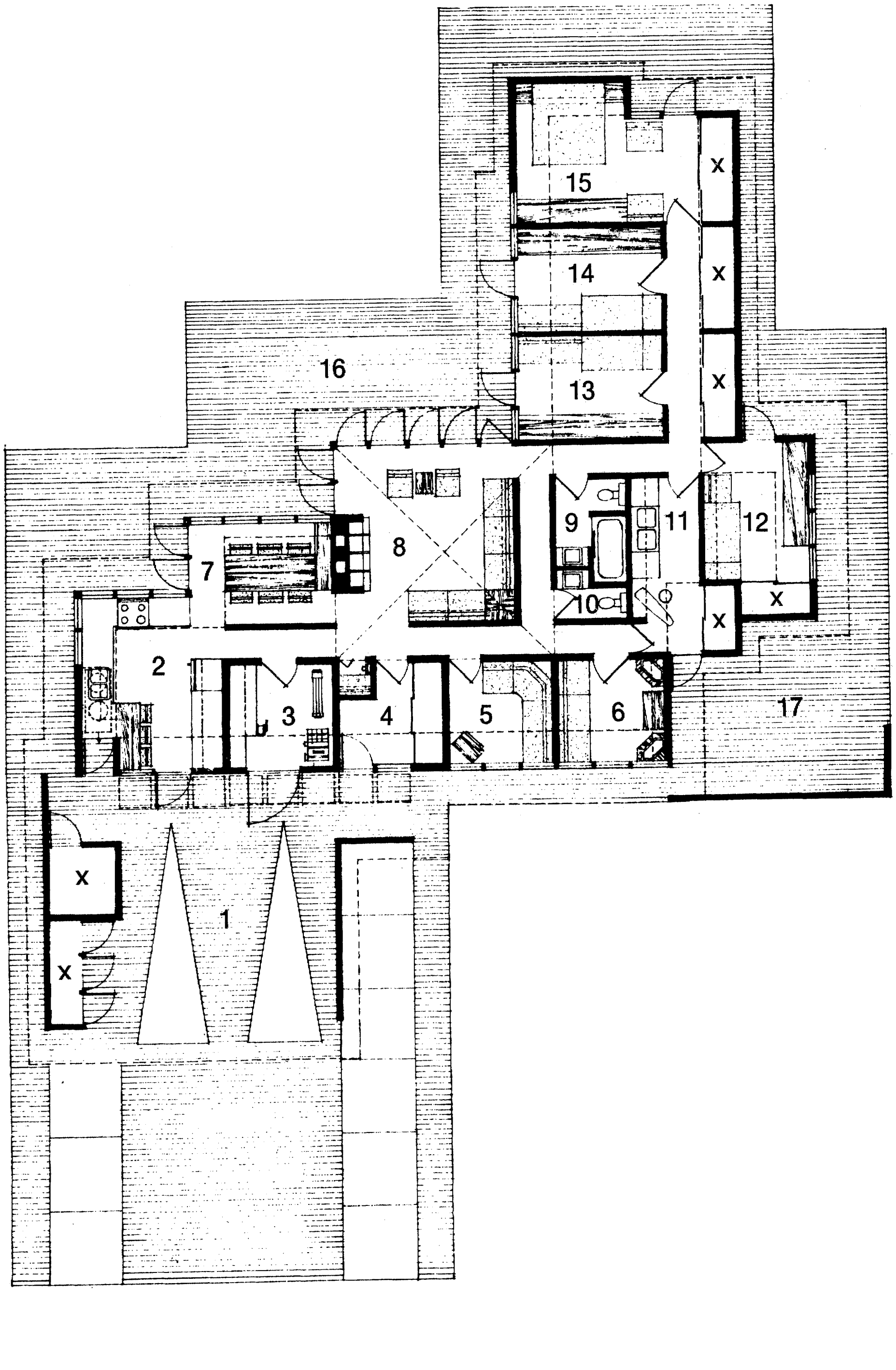




Above: Two computer drawings of different perspective views of the first stage of the house design. These preliminary views allowed the architect to pre-test the design and to correct areas which would have caused detailing difficulties.

Final rendering of the house design, based on the computer plot.





Ground plan:

1) Carport
2) Kitchen
3) Shop
4) Entry/storage
5) TV room
6) Sound room
7) Dining room
8) Living room
9) Bathroom
10) Washroom
11) Laundry, sewing, ironing room
12) Bed/study room
13) Bed/study room
14) Bed/study room
15) Master bedroom
16) Terrace
17) Drying yard
X) Storage

A design problem is by nature a lattice, that is, a structure with overlapping requirements. The fact is that in the conventional formmaking process the problem is not only based on artificial groupings of requirements, but is also considered to form a tree, with each requirement a member of a single "group," with none of them overlapping or interacting. This is a major weakness not corrected in the HIDECS 2 computer program and only potentially corrected in the programs called SIMPX and EQCLA. Yet, by manipulating the outputs of the SIMPX program and the BLDUP program and using them as data for further computer runs, a hierarchy based on a lattice may be produced.

The hierarchy yielded by this procedure is a formal statement of the design problem, and supports the contention that a problem clearly stated is half solved. But the procedure itself possesses no magical power. Its effectiveness depends on a thorough understanding of its use in the context of the entire design process. The designer must be able to provide a comprehensive set of misfit factors and their links, and to solve the hierarchy of subproblems in the specified order. The computer-aided analysis defines, but by no means shortens, the synthetic process. Such an analysis still demands unbiased, imaginative work by the form-giver.

This example shows how a formal picture of the design solution can be synthesized from a formal mathematical statement of the design problem. A recomposition of overlapping subsystems of the system of misfit factors and interactions into a hierarchy appears to be a valid model for the synthesis of form.

It is not suggested here that the process described will yield one perfect final form, but it is clear that a form which reflects a correctly synthesized formal solution will possess both unity and clarity of purpose. A single problem analyzed in this way might yield many such forms.

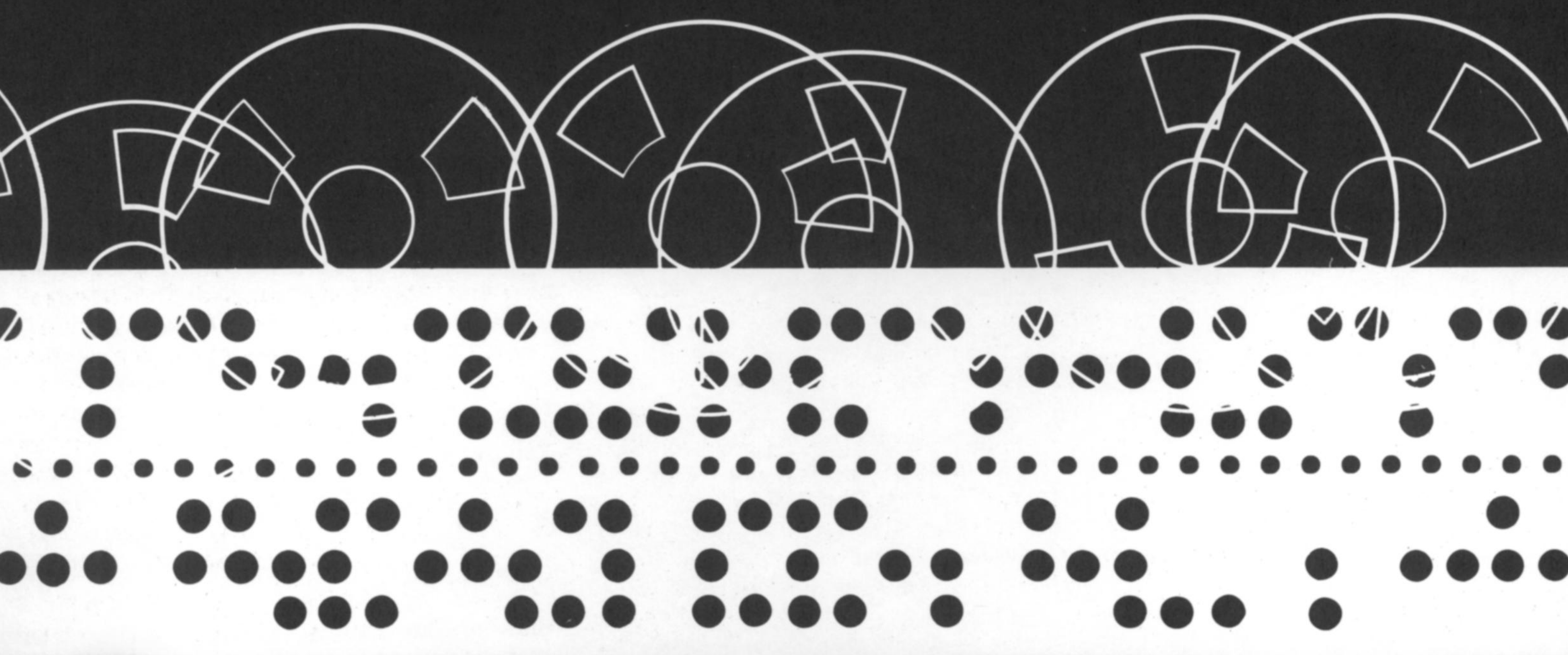
In addition to the above procedure, the designer has other opportunities to use the computer as a partner in improving our environment. For instance, the designer can, by using automatic drafting equipment, pre-test and evaluate his creations on a dry-run, pre-construction basis, as jet manufacturers already do today. This means that these built-in correction factors in the actual drawing stage will permit the designer to make every job a custom-tailored affair.

Even if we consider the computer simply as a high-powered drafting machine, this in itself is very significant. It will free creative architects for the high-level tasks for which their training ought to prepare them, rather than relegate them to reproducing volumes of repetitious drawings. In fact, the computer may enable man to pursue those roles which are uniquely human.

REFERENCES

1. Research for the article was financed by a National Research Council (Canada) Grant
2. Alexander, Christopher, *Notes on the Synthesis of Form* (Harvard University Press, Cambridge: 1964), pp. 60 - 70
3. Alexander, Christopher and Manheim, Marvin L., *HIDECS 2: A Computer Program for the Hierarchical Decomposition of a Set with an Associated Linear Graph* (Massachusetts Institute of Technology Press, Cambridge: 1962)
4. Alexander, Christopher, *HIDECS 3: Four Computer Programs for the Hierarchical Decomposition of Systems which have an Associated Linear Graph* (Massachusetts Institute of Technology Press, Cambridge: 1963)

Punched tape and reels of magnetic tape used in photo-typesetting equipment.



Kenneth G. Scheid is president of K. G. Scheid and Associates, Pittsburgh, and formerly head of the Graphic Arts Department at the Carnegie Institute of Technology. He is interested in design education and the fundamental changes which are occurring in the design field today.

Scheid points out that publishers, editors, graphic designers, printers and typesetters are constantly being influenced by current developments in computer technology. It is now technically possible to establish systems in which the process of editing, setting type and printing is fully computerized. Newspapers and large publishing companies are most interested in these developments. Some papers in the United States already use systems where the reporter types out his story on an electric typewriter which simultaneously produces a tape for automatic typesetting. The editor's corrections are merged with the reporter's tape and fed to a computer which controls a phototypesetter or a linotype machine. It is possible to produce systems where the editor sits in front of a television screen and calls up copy to appear on the screen, adds or deletes lines with an electronic stylus or light ray pen, then pushes a button causing the edited version of the copy to appear on the screen. He presses another button and the computer will have the type set and ready to print. Although this seems very futuristic, the elements for such systems are available, and as Mr. Scheid explains, work is being done which will change the functions and activities of the graphic designer, the typographer, the book designer and the layout artist.

In the very near future, graphic designers will have to concern themselves with these techniques, familiarize themselves with the technological possibilities and evaluate the aesthetic qualities involved.

COMPUTERS, PRINTING AND GRAPHIC DESIGN

by Kenneth G. Scheid

Printed communication can be achieved by means of an ever-widening variety of systems with modern technology. The most familiar one for capturing information in visual form is the typewriter, and the carbon copy process is very efficient in producing a limited number of useable duplicates. The camera is another common method especially useful for presenting nonverbal information, and photographic printing is a very efficient means of supplying a limited number of duplicates, provided that these copies are not needed instantaneously. A third familiar device is the printing press — ranging in scale from the office duplicator to massive newspaper and magazine units that permit virtually unlimited numbers of ink-on-paper copies to be produced rapidly and at low cost.

The selection of the most satisfactory system for a particular communications purpose and audience size, at any given stage of technological development, involves consideration of three major requirements:

1. Unit cost
2. Communications speed
3. Communications effectiveness

There is usually a trade-off among these factors. For example, a single letter will generally best be typed out on an office typewriter, at a unit cost of a dollar, in about fifteen minutes, with the design effectiveness that an individually-typed communication permits. However, when a letter should be copied a number of times, it will probably not be typed but run through a copying machine at an approximate cost of ten cents in less than a minute, because the communications effectiveness is entirely adequate compared to the cost of copy typing. Because of the fact that one might be willing to pay a slightly higher price for a graphically more effective result, a wide market has developed for the executive typewriter, which gives deep black, sharp and proportionally-spaced letter images. But no one is very likely to have the type for a single letter set and then printed, regardless of the quality of the result thus achieved.

To write the same letter to many persons, one would have to choose between typing each letter individually, duplicating an original and inserting personalized information, or typing automatically except for the manual insertion of the personalized material. The choice of process would be governed by the number of copies needed, equipment available, labor and time required, and the graphic effectiveness of each result.

Suppose, instead of a letter, a lengthy document which includes photographs and drawings — some even in color — has to be communicated to a large audience. Efficient printing systems for making thousands of copies would be offset lithography, gravure or letterpress printing. Here the type probably would be set in metal or by a photo-setting machine on film. The costs would be low — first, because the process permits the recording of far more information in a given area and thus economizes on paper, press time, binding time and mailing costs; second, the level of graphic effectiveness is highest; and third, despite a higher initial cost, the unit cost, when spread over thousands of copies, is comparatively low.

This example can be matched by numerous others. Cost, speed and design effectiveness govern the selection of systems for printing newspapers, journals and magazines of various circulations, advertising material, schedules, books, packages and business forms.

Rapid developments in computer technology have made improvements possible in the cost, speed and graphic effectiveness of printed communications. Some of the more important developments include the tape-punching typewriter, the tape-driven typewriter and typesetter, the magnetic or optical scanner, the high-speed chain printer, computerized typesetting employing the high-speed phototypesetter, computer-generated typesetting, versatile graphic arts film, the long-run offset plate,

Electronic retina computer reader produced by the Recognition Equipment Corporation, Dallas. This optical scanner reads the author's manuscript directly, eliminating the intermediate production of a punched tape. Reading systems of this type work with a speed of up to 2400 letters per second, and read various sizes of typewritten manuscript sheets.

the paper offset plate, the offset duplicator, the electrostatic duplicator, the color separation scanner, the web offset press, plus a whole series of improvements in paper, inks and other materials used in prinitng.

In selecting a printing system, one must integrate new technological developments into the reproduction process. Since that part of the process which precedes the production of multiple copies on a press is to a considerable extent an information storage, processing and conversion function, the computer becomes a possible new source for cost reduction, increased speed, and improved graphic effectiveness. The following are some of the functions which computers perform in printing systems.

INFORMATION RECORDING

Text and numerical data can be stored in the computer for subsequent processing and retrieval by means of a keyboard-punched paper tape or direct keyboarding, magnetic tape, punched cards or optically-scanned typescript. The most useful system to date has been the one which uses the keyboard early in the editorial-printing cycle by making use of typewriters that simultaneously produce a paper tape and printed proof copy or direct signals to the computer. (If the printed copy, usually called "hard copy," need not be exact, various "shorthand" codes for common words and operations can be used to reduce the number of keystrokes required to store the information in the computer.) Where the raw information is in voice form, a stenotype keyboard with a punch or signal-generating attachment might be used to convert the voice directly to input tape or input signals, in which case the computer must have the capability of translating all stenotype codes to natural text. This would require a dictionary memory unit which is not yet economical. Scanners, on the other hand, can read manuscripts typed in pre-selected faces, but their cost is very great and they may only be useful in cases where speed is vital, volume is massive, or the information does not require another keyboarding for optical reading.

The advantages of the punched tape or direct keyboarding are not only the relatively low capital costs, but also the fact that the keyboard can be used to re-type edited copy or multiple copies automatically, with operator intervention only when new information is inserted, in which case a new tape or signals directly to the computer can be simultaneously produced.

EDITING

The information stored in the computer can be edited in a number of ways. Perhaps the most direct approach is to use the original hard copy for making final author's alterations, editorial changes, and proofreading corrections, and then to keyboard only the corrections and merge them in the computer with the original information. This editing process can continue through whatever number of revisions is necessary without redundant keyboarding.

Other approaches include a) reading the output of a chain printer (a computer output device which prints alphabetical or numerical characters with a speed of up to 1000 lines per minute) and then keyboarding the corrections only; b) reading corrections in by optical scanning; and c) reading copy on a cathode ray tube display and making changes directly with a light pen, an electronic device for drawing or erasing.

REORGANIZATION

One of the most valuable functions performed by a computer occurs when the same set of information has to be printed in several different forms. For example, an alumni directory may list graduates alphabetically, then by class, then by location, then by type of degree and, finally, by occupational groups. With the com-

IBM Datatext computer system: This system allows up to 80 customers at different locations to use a single computer to type, edit and up-date text material.

Computer installation produced by the Digital Equipment Corporation, Maynard, Mass., and installed at the *Worcester Telegram and Gazette,* Worcester, Mass., for newspaper typesetting.

puter there is only one keyboarding for each alumnus needed because the various sortings can be performed automatically. Bibliographies, indexes, abstract searches, catalogues and many other reference works can be similarly processed. Where statistical data is to be presented in a variety of ways, recomputations and reorganization is likewise possible.

TYPOGRAPHY

The best-known use of the computer in printing is to perform a whole sequence of typographic composition functions that are normally done by layout or pasteup artists, keyboard operators, strippers and make-up men. One gives the computer a complete set of design instructions covering such matters as line, column and page measure; type faces and sizes to be used; indents, run-around, tabular formats and other variations from straight columnar form; limits in letter-spacing, word spacing and line spacing; hyphenation and justification desired, if any; rules concerning paragraph and page endings and the position of page numbers and footnotes.

Once these design instructions are established, the computer then makes the necessary calculations, hyphenates words by reference to rules of logic and a stored dictionary of exception words, and organizes the information in such a way that, on command, a fully programmed paper or magnetic tape is produced to drive a typesetting device. At the present time the advantage of the computer is greatest for setting columns of straight matter. Where hyphenation can be designed out of the format, or performed manually by an operator at the computer input/output keyboard, computer cost is reduced. While there are a number of programs that give a so-called area make-up which is used especially for display advertising, page make-up is less common but it is likely to grow as the technology of typesetting improves.

TYPESETTING SPEED AND VERSATILITY

Apart from its high speed in performing the typographic composition functions described above, the computer, as noted earlier, permits a substantial net reduction in the quantity of keystrokes required to produce printed communication, and also speeds up the keyboarding rate by eliminating the traditional requirement of the operator to make justification and hyphenation decisions as he proceeds. Because keyboarding is the most time-consuming operation in printing, these improvements can make substantial savings possible in the total printing costs. In addition, the operation of the new high-speed phototypesetters can best be controlled by a system of electronic circuitry that might be described as a special purpose computer. Such computers are needed to achieve the high speeds and also to operate film-setting devices which have the ability to produce a far wider range of type faces than is found in conventional manual machines.

NON-VERBAL IMAGES

Assuming that the composition unit is the page, film images of photographs, drawings, graphs, paintings, etc. have to be inserted by hand, which seriously slows down the printing cycle. This pictorial matter should also be processed through the computer, halftoned, and placed in position on the page simultaneously with the text matter. No such system is in operation today, but several are under development to be used for special publishing purposes, and at least one is being outlined for the production of a completely made-up newspaper page. The linking of an electronic color separation scanner to the system is being considered, so that even process color negatives could be produced and placed in position as well.

It is likely that this use of the computer will be most important where the non-verbal matter is created in the computer, as in the case of engineering drawings, charts, graphs and scientific symbols. For simple conversion of pictures and other outside source material to printing form, the computer may not at present be an economical substitute for the existing efficient graphic arts processes.
The computer offers some interesting economies where subsequent editions of a printed

communication are to be published. For example, if a book has been printed using a specific typographical arrangement, a later edition with different typography would normally need a completely new keyboarding. Not so with the computer, if a tape of the original information has been retained. It is then only a matter of processing this information in terms of the revised design and driving a typesetting machine automatically to produce the new typography.

The prospects for the use of computer technology in printing systems increases because of reductions in the cost of computers. These reductions appear in the form of lower-priced general purpose computers, the possibility of time sharing from central in-plant computers or from computer service bureaus, and the increasing availability of computer time for printing purposes at low costs on computers that users have initially installed for other purposes. These cost reductions continue to increase the computer's economic advantage over human operators in the performance of typographic composition and other functions.

PRESS SYSTEMS, PHOTOTYPESETTING AND COPYPROCESSING

Other technological developments that also add to the growing economic advantage of the computer are the following:

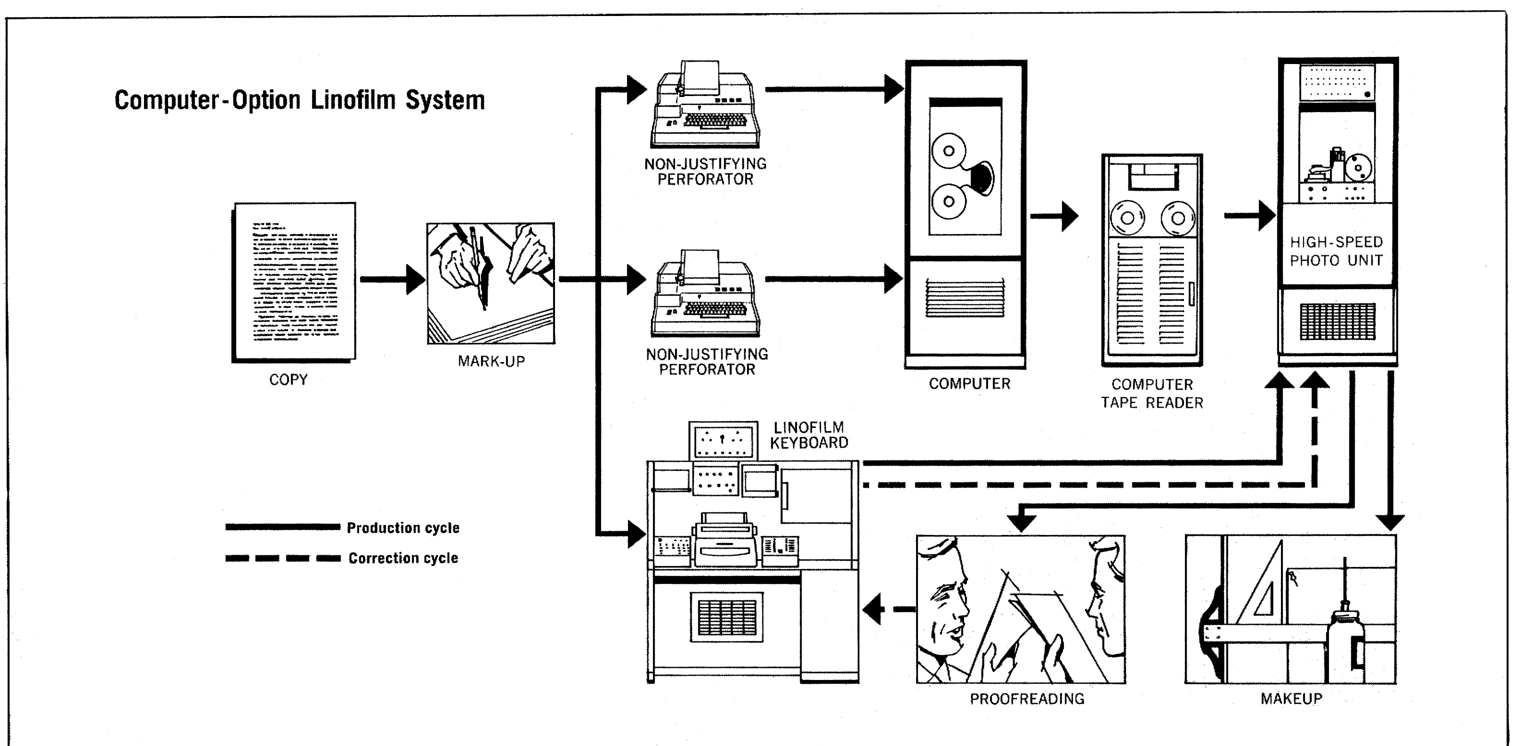
PRESS SYSTEMS

All of the important improvements on press systems such as speed, cost and quality, involve the use of printing plates or surfaces that are made from images stored on paper or film instead of metal. These improvements include the perfection of high-speed web offset printing of newspapers, magazines and books, the development of thin, photochemically-prepared letterpress plates, the continued improvements in rotogravure, and the probable emergence of a versatile electrostatic printing technology. For all these processes it is more logical to set the images directly on paper or film, rather than going through a metal casting and subsequent conversion process. Thus metal typography will tend to be replaced by phototypography.

PHOTOTYPESETTING

The first real breakthrough in typesetting speed since the invention of the linotype machine came with the perfection of the phototypesetter. As a result of this improvement, a phototypesetter in the same price range as a tape-operated linecaster can set twenty or more characters per second, 80 times faster than a tape-operated linecaster at only about 12 times the cost of such a linecaster. In contrast to that, a manually-operated linecaster can mix up to four type faces or type sizes and set only about three characters a second, or, if tape driven, about eight characters per second. Phototypesetters set a full page of a telephone directory in less than a minute. Phototypesetters with even faster speeds based on electronic character generation and capable of replacing the high-speed chain printer are under development. All these new systems must be operated by a paper- or magnetic-tape, and their real

Diagram showing production and correction cycle of the Linofilm computer system.



advantage becomes apparent when this tape is produced through computer processing.

COPY PROCESSING AND INPUTS

The use of the computer to edit and re-organize text in the pre-typesetting stage, as well as the development of editing typewriters which produce paper tapes, also adds to the value of the computer as a printing system. Computerization also eliminates redundant keyboarding by early inputting in the system. Moreover, since keyboarding will remain the most expensive operation in the process, the earlier it is done the more likely that it can be performed at a much lower cost by office typists than at a much higher expense by typesetters or production workers in the printing plant. Furthermore, the newer systems will reduce the cost and increase the speed of making corrections, since they will be made from hard copy at the outset under the more direct control of the publisher, rather than after the information has been typeset and must be galley read.

Other input processes, such as the optical scanner mentioned earlier, are also being developed to take advantage of the computer's capabilities. Most of them are geared to a situation where information is already stored on tape which then can be accepted directly by the computer without the usual keyboarding.

Computer-oriented systems are bound to have important consequences for publishing and printing markets, production costs, industrial structure and manpower requirements. It is not clear yet how these systems will alter the functions or processes of graphic design. This is not to say that designers can neglect becoming well-acquainted with this new technology, for there are aspects of it in which they must become involved. More importantly, it is an evolving technology with still-to-be-imagined usefulness to which the creative, inventive, knowledgeable designer may ultimately make vital contributions.

Already in the immediate future, the graphic designer may be asked to contribute to the evolution of these systems in these ways:

1. He may be requested by clients or employers to evaluate proposals to change from conventional printing processes to one of the computerized systems. In particular, his role may be to investigate and appraise possible gains or losses in design effectiveness that may accompany cost reductions and speed increases. One can hope that he will take the initiative in this involvement, for otherwise a stream of new, knowledgeable computer printing specialists may set the decision parameters to which his skills should be applied.

2. He may be far more intimately concerned in cases where his employer is seriously considering the installation of a computerized system. Indeed, given the "office," characteristics of these systems, there may be a considerable number of large corporations or other users of printed communication, as well as large publishers, who will choose to install their own partial or complete typesetting systems. Where this is the case, the designer will find it necessary to participate actively in selecting the computer equipment and developing programs and methods.

In either of the above instances, graphic designers will become involved in the making of typographic plans that in turn become central elements in the printing systems program. These include:

(a) Specifying what typographic composition and pictorial composition capabilities the system is to have, and which of them shall be automatically performed.

(b) Specifying the type faces and sizes that the system should be capable of outputting, and the mixing capabilities that are needed. In addition, the designer may want to prescribe minimum quality standards for computer-generated type characters.

(c) Determining whether the system can produce effectively if certain typographic conventions, such as hyphenation, justification, word spacing criteria, etc. are dispensed with.

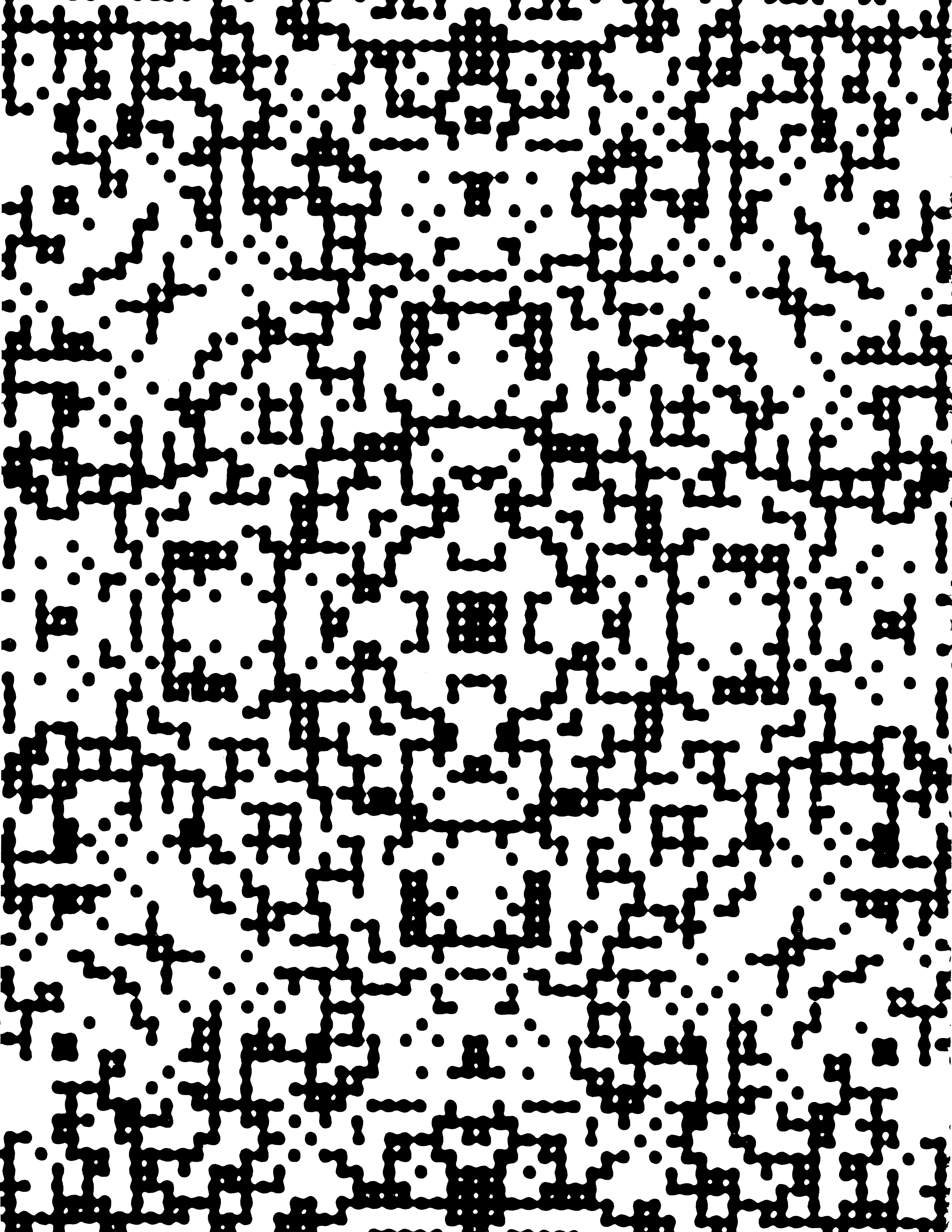
(d) Prescribing layout and page make-up systems, for example, using mathematically-described grids that permit easy, early planning and copyfitting, and that allow maximum use of automatic processing.

(e) Working experimentally with the light pen at a cathode ray tube interface with a system, to build into it a creative capability that may become its most interesting design aspect.

(f) Collaborating with systems designers and programmers to assure that design factors are fully considered in the most effective way.

The graphic designer may find that these new systems will substantially expand the need for design services in the production of printed communication. For example, if typewriter printing gives way to phototypesetting of much internal communication, including computer printout, it will certainly pay many users to employ design services in setting up these more effective systems.

These comments simply serve to open up a rational consideration of the computer's future role in graphic design. Indeed, this is all one can now accomplish: to alert the design profession and to encourage it to explore, whenever possible, the graphic potential of new technologies.



COMPUTER-GENERATED MOVIES, DESIGNS AND DIAGRAMS

by Kenneth C. Knowlton

Dr. Kenneth C. Knowlton works in Computer Programming Research at the Bell Telephone Laboratories at Murray Hill, New Jersey. He is interested in experimental "computer languages," and one of the "languages" he has created is capable of generating designs, diagrams and animated motion pictures. One problem which is difficult to solve in motion picture languages for computers is the production of different shades of gray. This problem has been solved by Dr. Knowlton in his BEFLIX language. Making animated movies by a computer programmed with BEFLIX has proven to be reasonable and feasible from a financial point of view. Moreover, BEFLIX can generate very complex patterns of considerable aesthetic merit which would be difficult and very time consuming to produce manually. The patterns sometimes come close to the products of oriental rug makers and thus provide the viewer with the impression that there may be a "human touch" to a machine-generated picture.

BEFLIX will be of great interest to the film maker who produces educational and other animated movies. The development of such languages together with the increased use of graphic output devices will make the computer very useful for the draftsman, artist and film maker.

An automatic microfilm recorder, composed of a camera and a cathode ray display tube which faces the camera, is now being used for the production of movies, designs and diagrams. Controlled by a computer, this recorder works at electronic speeds and can plot points and draw lines many times faster than a human draftsman.

One component of the automatic microfilm recorder, the cathode ray display tube, resembles a television tube except that it is controlled not by signals from a broadcasting station, but rather by signals from the electronic computer or a computer-written magnetic tape.

The microfilm recorder "understands" various kinds of commands. It "understands" the instruction to advance the film in the camera to the next frame. It responds when commanded to display a spot of a certain brightness at specified coordinates of a 1024-by-1024 grid on the tube face. It also draws a straight line segment from one such point to another. Some display units, such as the one produced by Stromberg-Carlson in San Diego, California,[2] in addition to producing graphic output, can generate typographic output by selecting, at high speed, characters from a given alphabet. This is achieved by deflecting an electron beam through the appropriate letter locations of an alphabet template in the tube. By passing through the template, the beam takes on the form of the letter and is subsequently deflected into a precise position on the display face of the tube. In some other display units, characters may be "drawn" by automatically plotting appropriate patterns of spots or line segments on the face of the tube with the electron beam.

Despite the simple principle of this machine's elementary operations, the machine can compose complicated pictures or series of pictures from a sufficiently large number of appropriately placed points, lines and characters. It can draw and film these elements fast enough to produce in a reasonable time a very complicated picture or series of such pictures. Current speeds of microfilm recorders lie in the range of 10,000 to 100,000 points, lines, or characters per second. This is fast enough to compose, in a matter of seconds, a television-quality image consisting of a fine mosaic of closely-spaced spots, or fast enough to produce several simple line drawings per second.

THE QUESTION OF EFFICIENCY

It should be understood at this point that the basic question in this research area is not whether a computer equipped with a microfilm recorder can produce complicated or interesting pictures. Indeed, almost any picture imaginable can in principle be composed as a particular configuration of points and lines. The basic question is whether this is an efficient and feasible way to produce useful pictures.

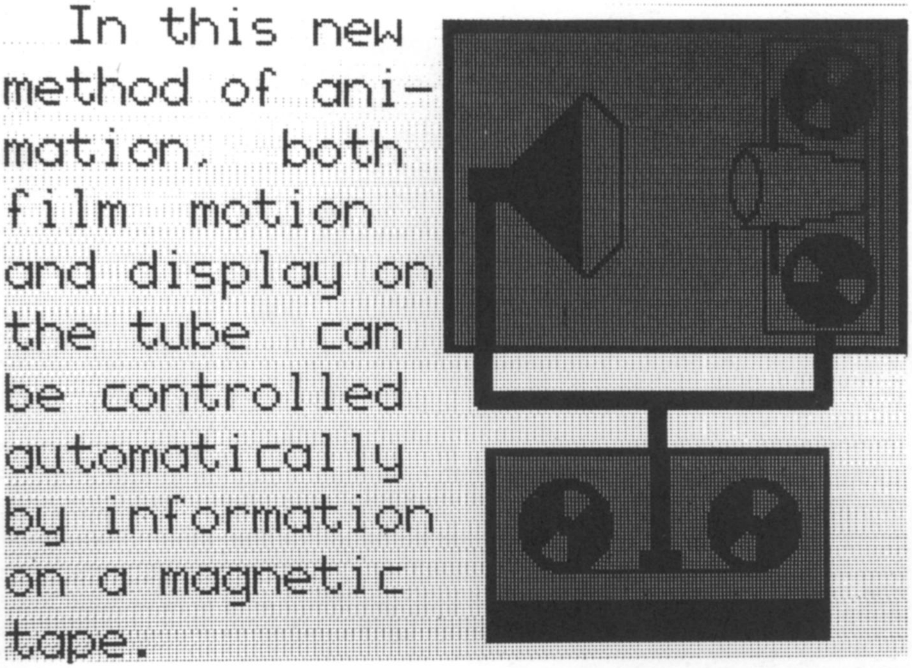
What the picture or movie producer would like to do, instead of specifying his pictures in terms of their elementary components such as spots and lines, is to talk in a more powerful language to the machine — the way he is used to talking to his human "assistants," or even more effectively. He would, for example, like to say: "Type such-and-such a title, center each line, give the letters shadows, then shoot 150 frames." The job of the computer, equipped with a program for understanding this kind of talk, would be to deduce, from a few statements in this powerful language, the very large number of corresponding instructions for the microfilm recorder, and thus to produce the pictures automatically.

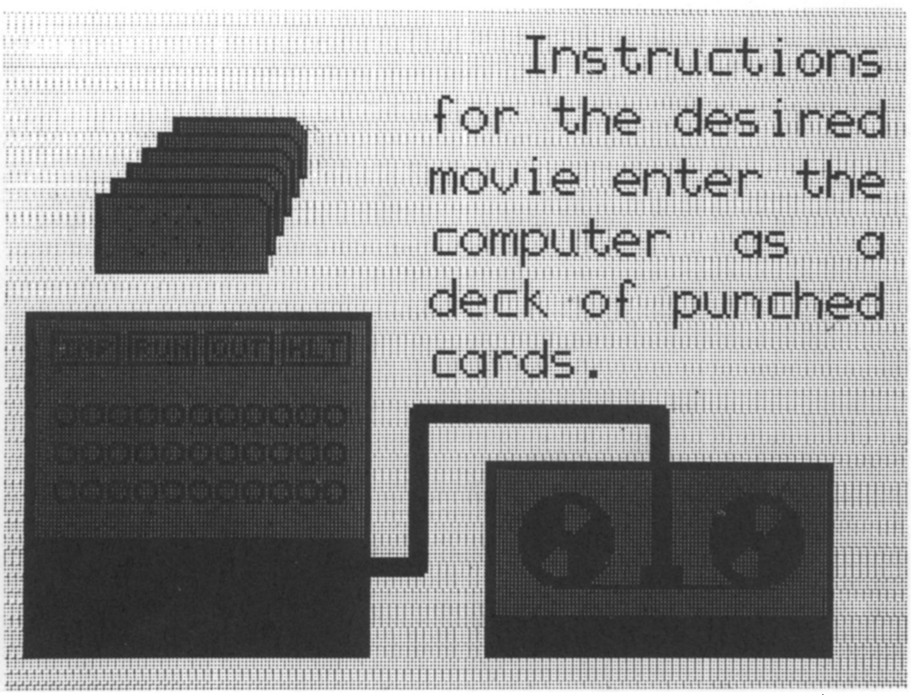
The interesting research questions are, first: "Is it possible to provide a sufficiently complete language so that most things which an animator or diagrammer wants to do can be expressed easily?" and secondly: "Will the computer's subsequent calculations be reasonably fast and hence economical?" From my experience with a particular "movie language," the answers are definitely "Yes" — at least for a broad category of pictures which I will herein describe.

THE BEFLIX MOVIE LANGUAGE

The work in this area has involved the mosaic type of picture, consisting of tens of thousands of spots or tiny characters, as contrasted with the line drawings of A. M. Noll and F. W. Sinden, also working at the Bell Telephone Laboratories, where a single picture is composed of hundreds of line segments. In particular, my contribution was the development of a special

Opposite page: Enlarged segment of computer-produced design.





Above: Frames from computer-produced film, "A Computer Technique for the Production of Animated Movies," produced by programming in the BEFLIX language. Each frame is a 252 by 184 array of spots (actually alphanumeric characters). Shown here is the microfilm printer, consisting essentially of cathode ray tube and camera, both controlled by information on magnetic tape.

The job of the computer is to accept instructions for the movie on punched cards and to compute and write spot-by-spot descriptions of pictures on this tape.

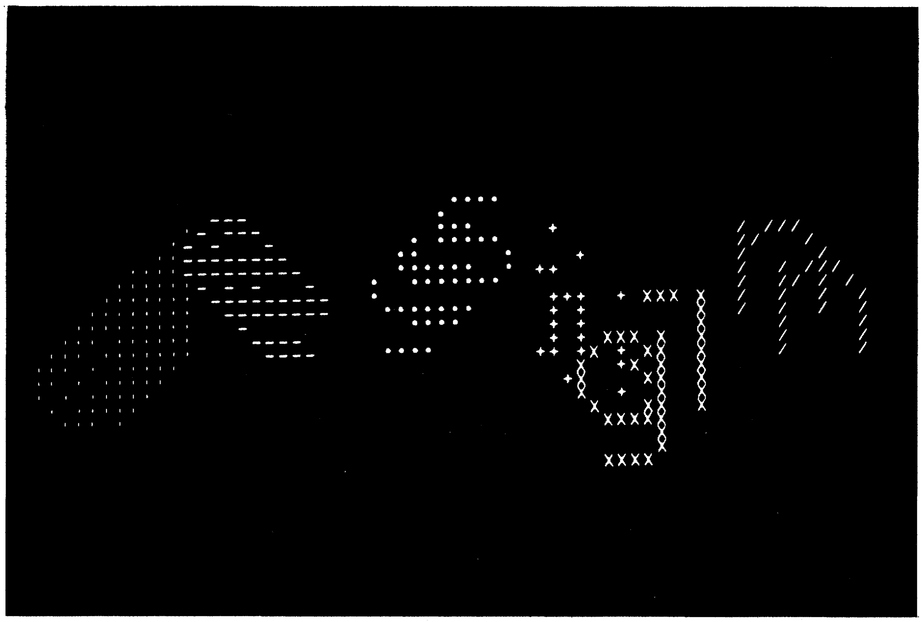
programming language. A programming language is used to prepare computer programs and, as such, it is a language that interprets human language to the machine. This programming language was geared for producing animated movies and, necessarily, a corresponding computer program, which "understands" this language and carries out the designated operations, was written.[3]

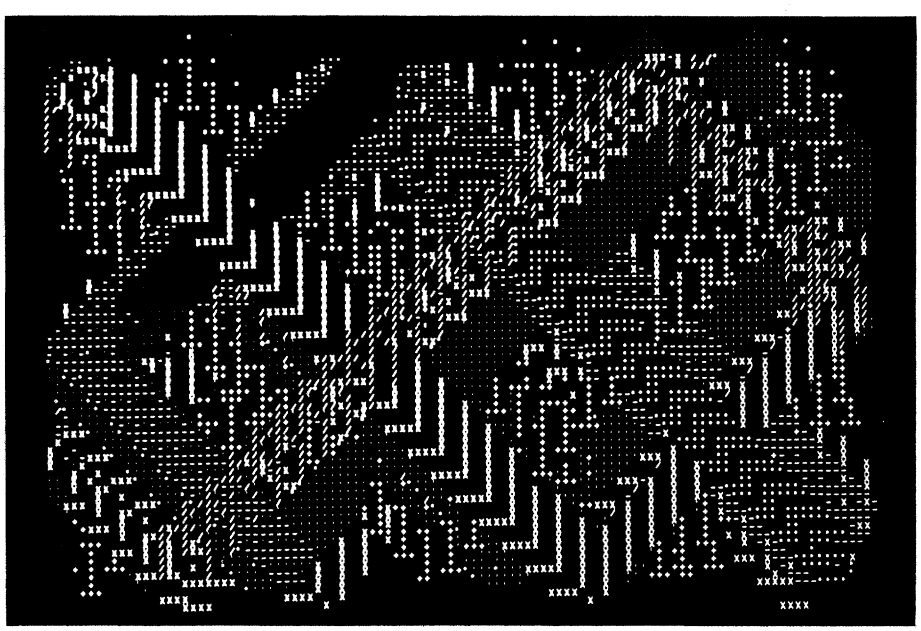
The programming language called BEFLIX (for "Bell Flicks"), deals with pictures as a large rectangular surface made up of 252-by-184 elements. Each of the 46,368 elements is represented within the computer by a number from 0 to 7, indicating the intensity of light at that point (which is equivalent to a "shade of gray"). Pictures are built up and modified within the computer by appropriate manipulation of these numbers. At desired times these numbers are used to direct the microfilm recorder in displaying the entire surface of 252 x 184 = 46,368 spots on the cathode ray tube in order to expose one frame of film. (Actually, these "spots" were generated by typing tiny letters, using the Stromberg-Carlson 4020 "charactron" tube in the "typewriter" mode — different characters being used for different intensities according to their overall gray level.)

By programming this language, I made a movie about the very process by which the film was made.[4] Many of the scenes, incidently, depict schematically the physical equipment involved. All scenes were produced by BEFLIX instructions such as those for drawing straight lines (which must, of course, conform to the underlying pattern and thus may involve minute steps), drawing arcs and other curves, "painting" an area with a solid shade of gray, copying contents of one area onto another, or shifting the contents of an area up, down, right, or left a specified number of positions. There are also operations for automatically filling an area that has been outlined by a specific shade of gray, for enlarging all or part of a picture, and for gradually dissolving one picture into another which has been drawn on an auxiliary "drawing board" within the computer.

In all, the language contains about twenty-five kinds of instructions, each of which is punched on an IBM card with appropriate parameters specifying just where and how the operation is to be performed and how many movie frames are to be produced at intermediate stages of the operation. For example, the instruction for drawing a straight line requires the programmer to specify beginning and end points, width of line in units, shade of gray, and the speed at which the line is to be drawn, expressed as the number of units the line should advance between successive frames of the movie.

The BEFLIX language actually does not do much which cannot be done by normal methods.





Three frames showing the development of a complicated design produced by giving the computer rather simple rules. BEFLIX was used in the "coarse" mode, where one picture contains only 92 lines of 126 characters each.

Top left: The computer begins by typing the word "design" using different characters as shown.

Center left: Letters follow trajectories, each writing its pattern in its own particular sequence.

Bottom left: After bouncing about inside an imaginary box, complex composite patterns form.

In many cases, however, drawings can be made with far less human effort than when drawn manually, especially drawings which exhibit symmetries or periodical repetitions of equal elements. In one scene of the film, for example, a computer console with thirty-three lights in three rows, is shown. These lights were produced by giving the computer instructions for drawing only one of them and the positions on the picture surface at which a light was to be located. In fact, it was not even necessary to give an explicit list of positions to the computer; only the rules for enumerating these positions were required.

This film, produced entirely by BEFLIX programming, presents a means of judging the economic feasibility of producing movies this way. It is a 17-minute film and thus about 25,000 frames long. However, some of the frames are identical and so there are only 3000 unique pictures. These 3000 pictures were actually produced by approximately 2000 lines (or punched cards) of BEFLIX programming, which took me two months as the sole producer-programmer. The other major expense was four hours of computer time on an IBM 7094 computer (two hours to check out the program and two hours for "production" run).

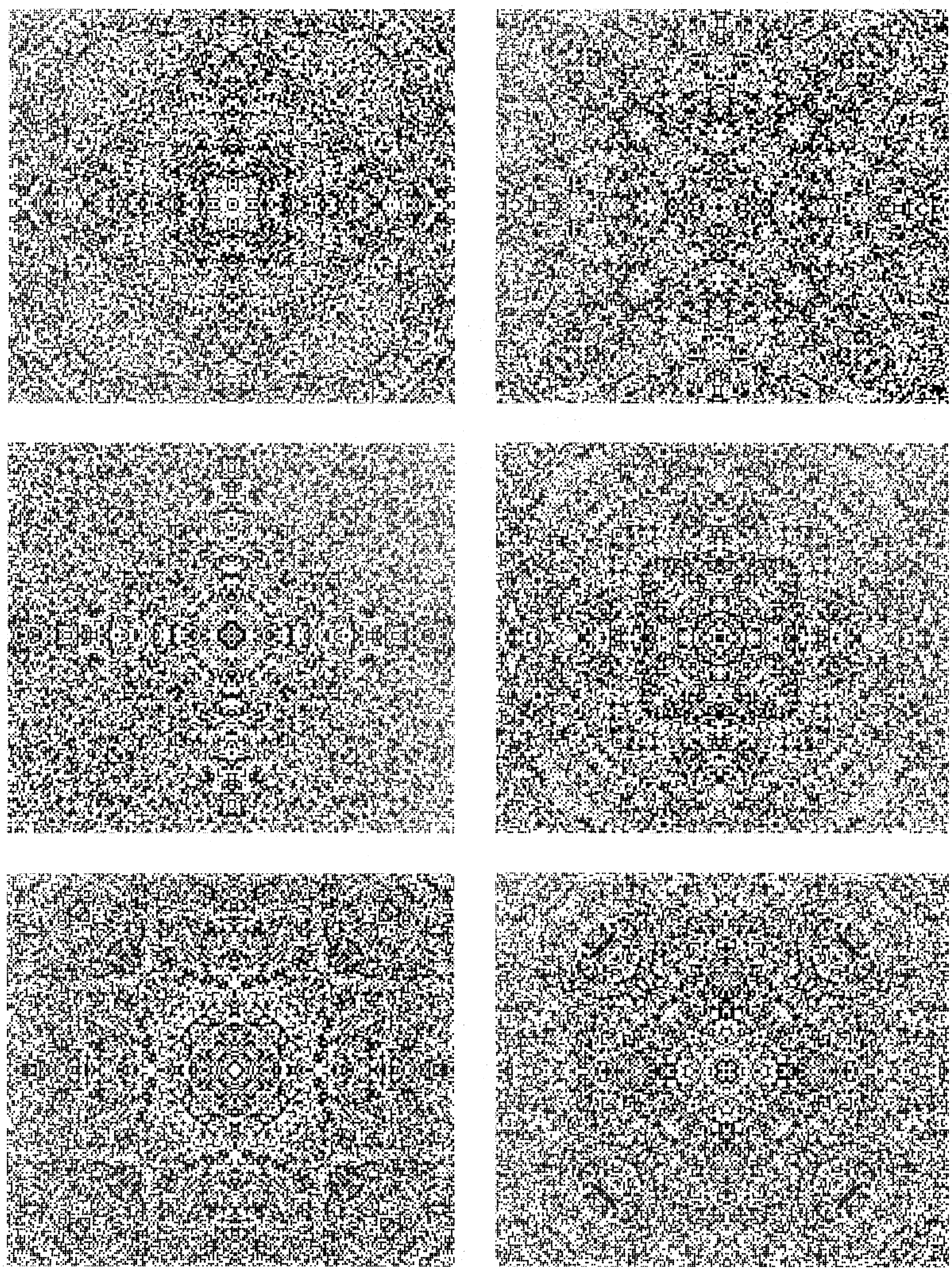
Other incidental costs bring the total cost up to $600 per minute of film which already falls at the lower edge of the range for manual animation of this quality and complexity. Costs of computer animation will undoubtedly go down with increases in size and speed of computers, with development of additional special-purpose equipment better suited to movie-making, and with further development of computer programming languages for the purpose. With efficiency improved only slightly from what is now available, it will also be feasible to improve the resolution of pictures and to do animation in color, both of which will require slightly more computation but only a minor increase, if any, in programming effort.

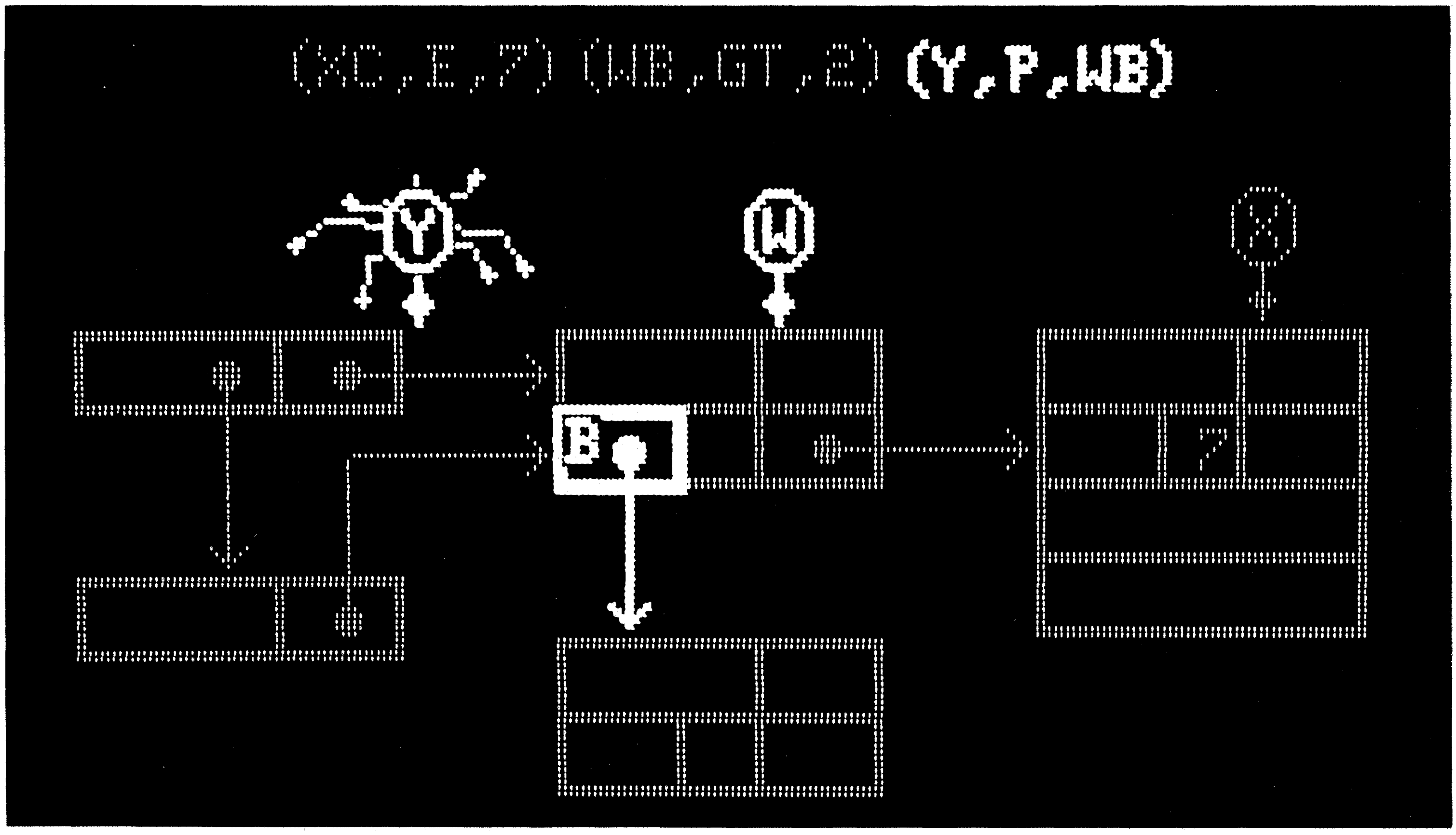
COMPLICATED PATTERNS FROM SIMPLE RULES

One should not immediately conclude, however, that all drawings and animated movies will soon be done by computer. In fact, it is difficult to imagine at this point how we would formalize for the computer, rules for drawing familiar cartoon characters. Instead, the computer will be used for the more schematic and geometric form and for those patterns and designs which are logically simple, in the sense that the rules for constructing them are easy to write in a programming language.

An example of such a "logically simple" design may look very intricate. I produced a visually complex picture by instructing the computer first to display the word "design," making up each letter of the word by a different typewriter character. Then the computer was asked to repeat each composite letter along a 45-degree trajectory following its own sequence of writing and/or erasing its pattern. These trajectories were limited by the walls of an imaginary box from which the letters bounded off until a fairly complex visual result was obtained. (For this series of pictures, the "coarse" mode of BEFLIX was used in which pictures are composed in a surface of only 92-by-126 elements.)

Designs exhibiting 8-fold kaleidoscopic symmetry, produced by Drs. Bosche and Julesz (Bell Telephone Laboratories). The computer is programmed to fill an octant at random and copy it seven times through appropriate reflections.





Frame from BEFLIX-produced film "L^6 — Bell Telephone Laboratories Low-Level Linked List Language." This narrated film explains the workings of a new computer language by means of moving "bugs," block diagrams and letters.

Further examples of complicated pictures produced by simple rules were made by Drs. Carol Bosche and Bela Julesz of Bell Telephone Laboratories, who added a pseudo-random number generator to the BEFLIX system in order to fill one eighth of each frame with black and white spots at random. The operations of BEFLIX were then used to copy this area so as to produce a pattern of eight-fold kaleidoscopic symmetry.

Finally, as another example of the use of BEFLIX for educational purposes, two films about a new computer programming language called L^6 have been produced.[5] These narrated films, totaling 48 minutes, were produced principally by myself in about four months' time, using ten hours of time on an IBM 7094 computer for a total cost of less than $500 per minute of film.

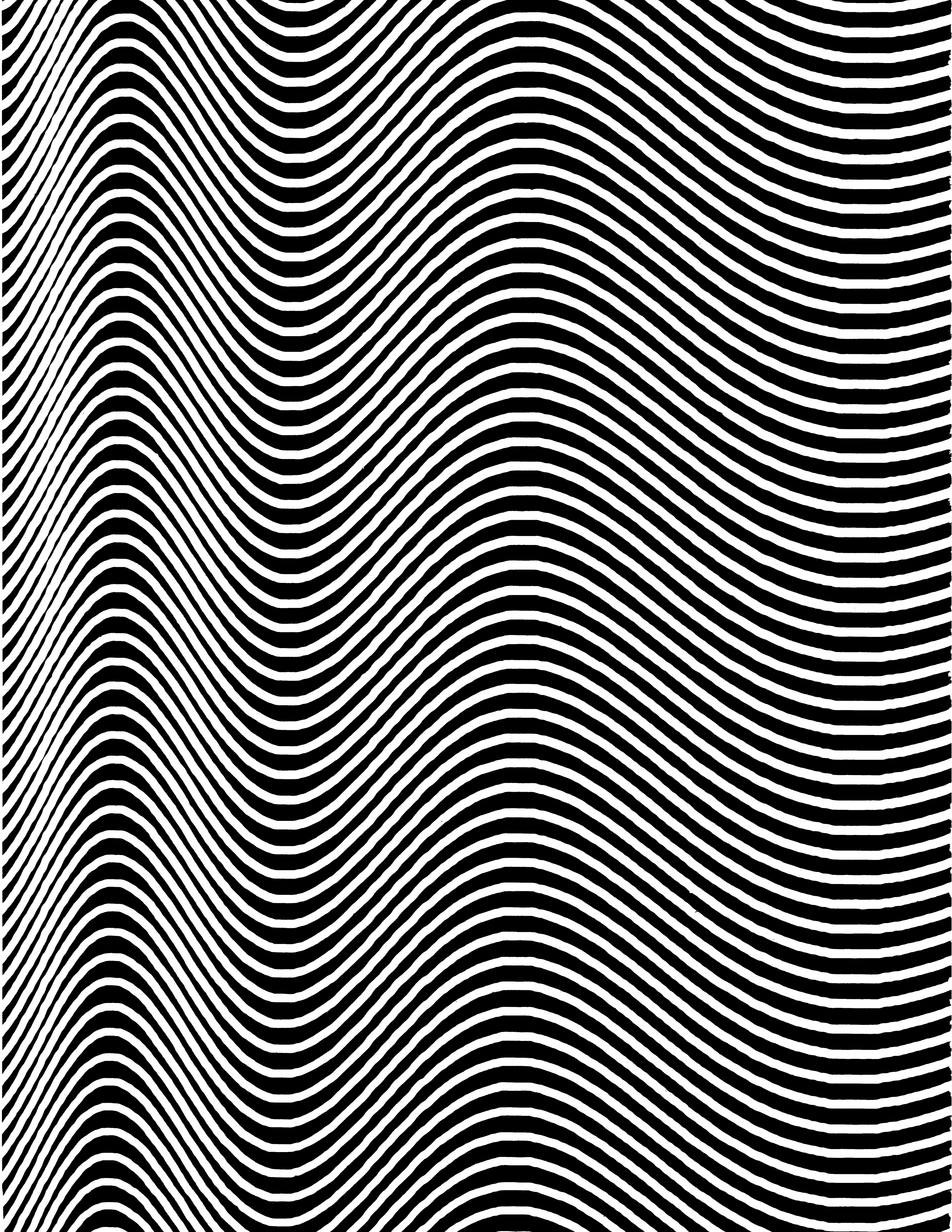
The L^6 language should be of great interest to designers who are developing computer-graphics systems for use in their work.[6] It is a programming language especially useful for certain types of non-numeric computation as it may occur in the simulation of circuits or the drawing of flow charts. In many cases it performs ten times faster than languages which have been previously used for these purposes.

CONCLUSION

BEFLIX and L^6 are cited as two languages which may be used for the computer production of animated movies, designs and diagrams. Further developments in programming languages, aimed either at more general usefulness or at more specific areas, may make the computer still more useful to the draftsman, artist, designer or architect. In addition, future developments in hardware, such as larger and faster computers, and graphics facilities permitting close interaction between designer and computer, should be expected to make the computer an almost indispensable tool in these areas.

REFERENCES

1. Knowlton, Kenneth C., "Computer-Produced Movies," Science 150 (November 26, 1965), pp. 1116-1120.
2. S-C 4020 High Speed Microfilm Recorder, Product Specification 281001-241A, September 8, 1960, Stromberg-Carlson, San Diego, California.
3. Knowlton, Kenneth C., "A Computer Technique for Producing Animated Movies," AFIPS Conference Proceedings 25 (1964 Spring Joint Computer Conference, Washington, D. C.) pp. 67-87. Appendices constitute a programmer's manual for the BEFLIX movie language.
4. "A Computer Technique for the Production of Animated Movies," 17 minute, 16 mm black-and-white silent film about the BEFLIX language and system, and produced by it. Available on loan from the Technical Information Libraries, Bell Telephone Laboratories, Incorporated, Murray Hill, N. J.
5. "L^6 — Bell Telephone Laboratories Low-Level List Language," 16 minutes, and "L^6 — Part II: An Example of L^6 Programming," 32 minutes. Both are 16 mm black-and-white films, with narration, computer-produced by BEFLIX programming. Also available on loan from Bell Telephone Laboratories.
6. Knowlton, Kenneth C., "A Programmer's Description of L^6," Communications of the ACM 9, No. 8 (August, 1966), pp. 616-625.



COMPUTERS AND THE VISUAL ARTS

by A. Michael Noll

Opposite page: Enlarged segment of computer-produced picture.

A. Michael Noll is on the research staff of the Bell Telephone Laboratories in Murray Hill, New Jersey. He has worked in the field of two- and three-dimensional films and moving pictures generated by computer to display scientific information, and is interested in the computer as a creative medium in the arts. His computer-generated pictures have been exhibited at the Howard Wise Gallery, New York. As a logical extension of Mr. Noll's computer technique for generating two-dimensional pictures, it became possible to use computer-generated stereoscopic drawings to achieve a three-dimensional effect if viewed with polaroid glasses. From the three-dimensional still picture, the next logical step was to animate the picture and then to produce moving, three-dimensional objects or kinetic sculptures.

The use of computers in communication research has uncovered techniques which can be applied to other areas of the visual and performing arts. For example, the same technique may be used to simulate or record choreography for dance. New computer languages are being developed which will make it very easy for the artist to communicate instructions to the machine. Computers with display equipment will also become more readily accessible with the net result that many more artists and designers will become computer oriented.

The techniques and concepts which have been developed during the past few years for using computers to generate visual displays of scientific data can be applied with very little effort to the production of artistic visual displays. Previous experience has indicated that emotional feelings and prejudices often against the concept of computer art exist on the part of both artists and scientists. Actually, what artists may learn in using these new computer techniques could be valuable to scientists and engineers. Conversely, artists may discover that computers are very advantageous in many artistic endeavors. Thus, computers may play an important role in linking art and science, with the artist and scientist mutually assisting each other. But before all this can happen, the artist and scientist must become familiar enough with the other's field so that each can recognize the advantages of mutual cooperation.

Art has always depended upon science and technology to supply both the medium in which the work is done and the tools for doing it. The techniques are common whether the computer is used to generate visual displays of scientific data (e.g., shapes and motions of mechanical systems, mathematical rotations of n-dimensional objects, motions of atoms in a fluid) or shapes and motions which may be important in design or as an artistic medium.

COMPUTERS AND PICTURES

All digital computers are similar in that they manipulate binary numbers under the control of a stored set of instructions called a program. The actual manipulation is performed by electronic circuits and devices. Most computers consist of a memory, an arithmetic processor, and input and output equipment. The program is stored in the computer's memory and instructs the computer to accept input data, to perform arithmetical, logical, and organizational operations with this data, and finally to generate some form of output.

As an example, the stored program might first instruct the machine to read the numbers punched on a card into a portion of the memory allocated for data storage. The arithmetic processor might then fetch these numbers from the memory and add them all together. The final sum might be returned to memory and from there printed out by a typewriter. Each operation is controlled in sequence by the stored program.

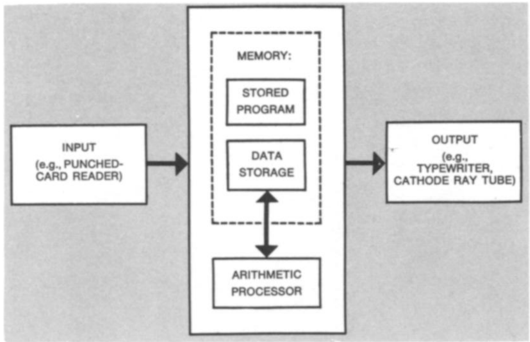
Computers are only capable of performing operations for which they have been programmed and instructed to perform. However, these machines are incomprehensibly fast in doing their tasks and are also capable of analyzing many factors to make different decisions. This extreme speed coupled with the inherent accuracy of digital calculation explains the desirability of computers.

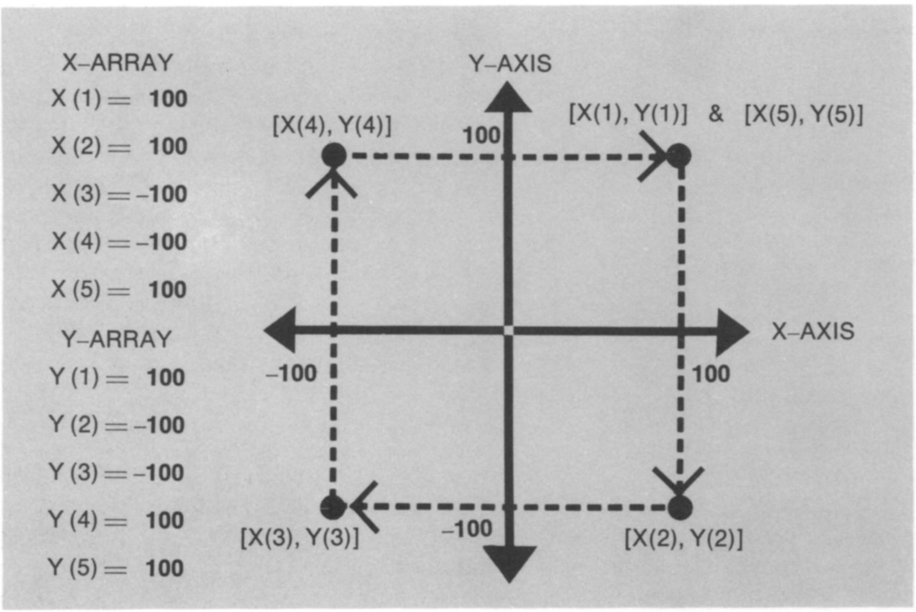
Recently, progress has been made in obtaining new output devices which allow the computer to produce visual output. One such device is the microfilm plotter.

The microfilm plotter consists of an interconnected cathode ray tube (conceptually similar to the picture tube in a television set), a camera, and some electronic control equipment. The cathode ray tube produces pictures on a phosphorescent screen with an electron beam which is electrically deflected across the screen to generate the desired picture. The camera photographs the face of the cathode ray tube. The required signals for deflecting the electron beam and for advancing the film come from the electronic control equipment, which includes circuitry for decoding instructions given to the microfilm plotter by the main computer. These instructions are commands for drawing straight lines between numerically specified points. In this way, pictures are generated on the microfilm plotter by instructions comprising a part of a program in the main computer.

As an example, suppose a square is to be generated on the microfilm plotter. The required program would first make two arrays of numbers whose elements are the X and Y coordinates of the corners of a square; i.e., $X(1) = 100$, $Y(1) = 100$, $X(2) = 100$, $Y(2) = -100$, $X(3) = -100$, $Y(3) = -100$, $X(4) = -100$, $Y(4) = 100$, $X(5) = 100$, $Y(5) = 100$. (The fifth point is required to close the square.) The program would, secondly, contain an instruction to advance the film to an un-

Diagram showing the basic concept for using the computer to produce typewritten or visual output.





Example of drawing a square with the microfilm plotter: Four points are located, each with a value of 100, in a quadrant of the X-Y coordinate system. An instruction to the microfilm plotter to connect these points would thus produce a square.

exposed frame. The next instruction would tell the microfilm plotter to draw a line between the five points whose coordinates are contained in the X and Y arrays. Thus, the point (X(1),Y(1)) would be connected to (X(2),Y(2)), (X(2),Y(2)) would be connected to (X(3),Y(3)), and so on. The last instruction would once again advance the film.

The arrays are blocks of sequential locations in the computer's memory that have been specified by the programmer as being associated with some name. For instance, in the above case the X and Y arrays each contain five storage locations where X(1) refers to the first location which contains the number 100.

In the preceding example of a square, the coordinates are all known and are therefore put into the X and Y arrays as constants. For more complicated pictures, the program might include instructions for mathematically calculating these coordinates. The essence of all this is that a picture can be specified numerically as a set of points which can be obtained as results of numeric calculations in a computer program. For complicated pictures, the only additional requirements would be more points and smaller line segments.

RANDOMNESS

A sequence of numbers would be described as random if an observer were unable to determine a formula for exactly predicting each number in the sequence. These numbers might represent the coordinates of points in a computer-generated picture. However, a practical picture cannot be infinite in size, and therefore some limit or range must be placed upon the set of permissible random numbers. For example, only random numbers between −67 and +240 might be allowed. Within a permissible set, certain numbers might occur more often than others, or alternatively the occurrence of all the numbers might be equally probable.

If all the random numbers in a sequence fall between the limits *a* and *b*, where *b* is greater than *a*, and if the occurrence of all these numbers is equally probable, then the numbers are said to have a uniform probability density. Such a sequence is specified by only the limits *a* and *b*. The occurrence of any number within these limits is just as probable as the occurrence of any other number.

Aside from the uniform probability density there is another density of considerable importance. It is called the normal or Gaussian density (after the mathematician Gauss who first formulated it mathematically). For a Gaussian sequence of random numbers, the numbers tend to cluster about an average. The larger the number compared with this average, the less probable is its occurrence. Sometimes the Gaussian sequence is "truncated" so that numbers much larger than the average are not allowed. The Gaussian density is also characterized by its standard deviation which is a measure of the spread of the random numbers about the average. For a very long sequence of Gaussian random numbers, 68.3 percent of the numbers fall within plus or minus one standard deviation of the average, 95.5 percent within two standard deviations, and 99.7 percent within three standard deviations.

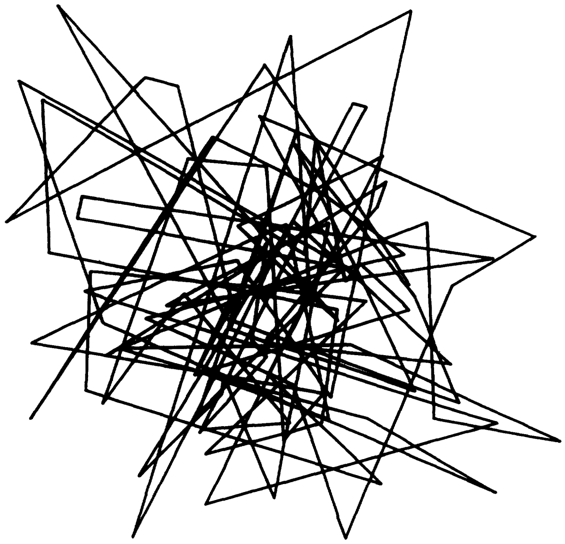
Numerical techniques developed for scientific and engineering use are available for generating a sequence of random numbers with either a uniform or Gaussian density. These techniques have been incorporated into special computer programs called subroutines which calculate the specified random sequence.

TWO-DIMENSIONAL PICTURES

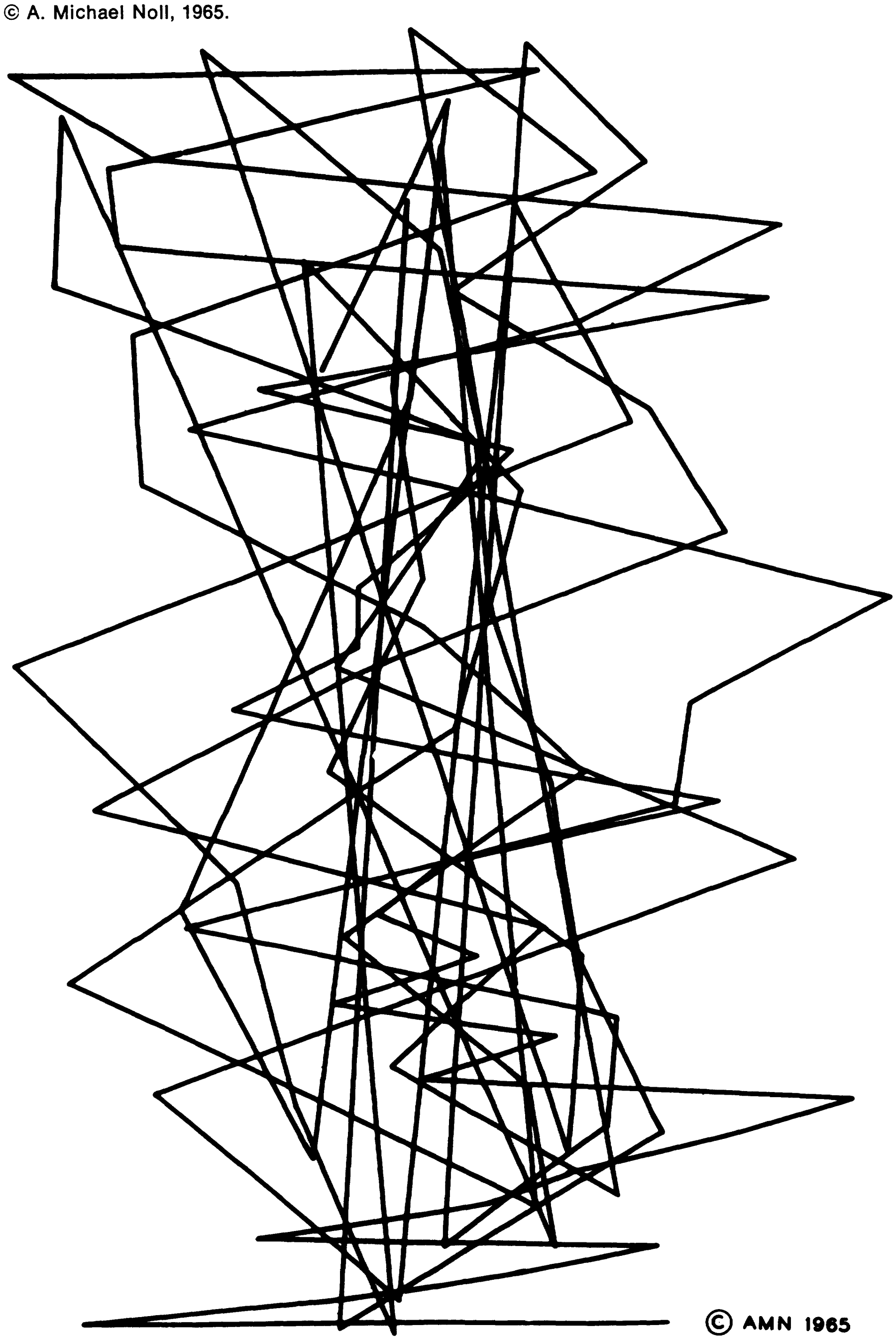
A computer program can be written instructing the machine to compute coordinates of points which, when connected together with straight lines, produce a picture.

Some interesting pictures can be generated with the computer by programming different random elements into the picture. In the first attempts along these lines, the random number subroutines for uniform and Gaussian densities

Picture with points determined by a random process and then connected by the microfilm plotter.



"Gaussian-Quadratic,"
© A. Michael Noll, 1965.



were used to generate arrays of numbers which became the coordinates of points. These points were then sequentially connected together with straight lines. The lines in the Gaussian picture cluster about the center since the average of the random coordinates was chosen to be the center of the picture. If desired, the computer could generate series of pictures in which the number of lines or the standard deviation was varied. The best picture could then be chosen from the series.

In general, completely random two-dimensional pictures are not very interesting. However, the computer is also able to mix together randomness and order in mathematically specified proportions to achieve a desired effect. The initial attempts at such mixing used Gaussian randomness for the X-axis coordinates but introduced a specified and non-random mathematical function for generating the Y-axis coordinates. A particularly good example of this mixing approach is shown in the picture, "Gaussian Quadratic". Ninety-nine lines join together 100 points whose horizontal positions are Gaussian. The vertical positions increase quadratically, i.e., the first point has a vertical position from the bottom of the picture given by $1^2 + 5x1$, the second point $2^2 + 5x2$, the third point $3^2 + 5x3$, etc. The maximum picture size is limited to 1024 units wide by 1024 units high, and thus the 30th point would be off the top of the picture ($30^2 + 5x30 = 1050$). To prevent this from happening, the vertical positions at the top are reflected to the bottom of the picture and then continue to rise. The result is a line that starts at the bottom of the picture and randomly zigzags to the top in continually increasing steps; at the top the line is "translated" to the bottom to continue its rise. The standard deviation of the Gaussian density is 150.

The exact proportions of "Gaussian Quadratic" were arrived at in a process of trial and error. The computer very rapidly produced series of pictures in which the different factors were uniformly changed. In this manner it became possible to bring an intuitive feeling for the pictorial effects of these factors into play.

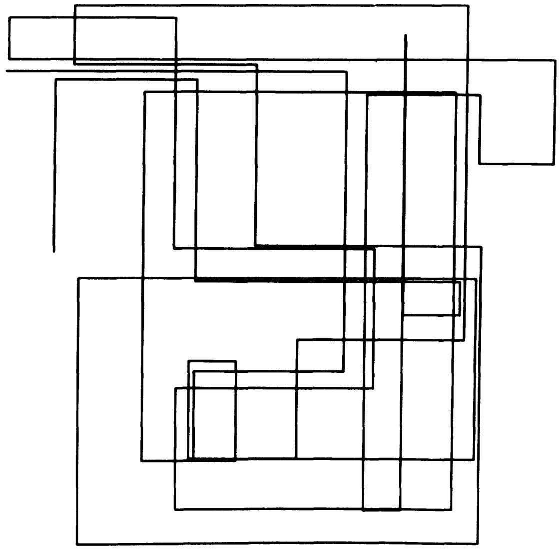
"Vertical-Horizontal" pictures were generated by a scheme in which only one of the two coordinates was changed (alternatingly) from one point to the next. The coordinates were otherwise random with a uniform probability density. "Vertical-Horizontal No. 1" consists of 50 lines with equal ranges in both directions; the number of lines in "Vertical-Horizontal No. 2" was increased to 300. "Vertical-Horizontal No. 3" consists of 100 lines with a range of −200 to +200 along the X-axis and a range of −500 to +500 along the Y-axis.

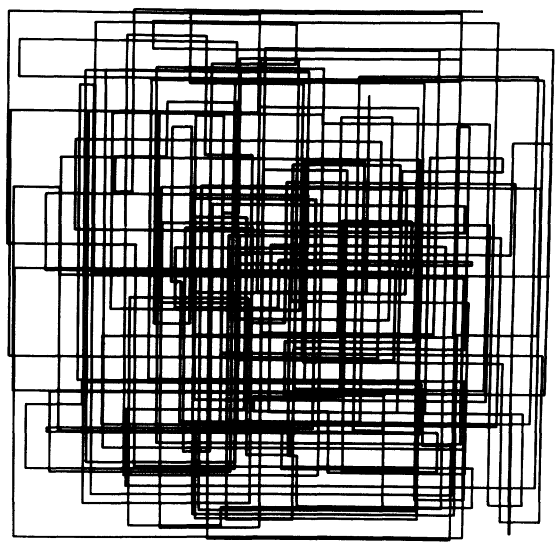
The preceding examples indicate that the computer in association with some method for obtaining visual output can be used to generate

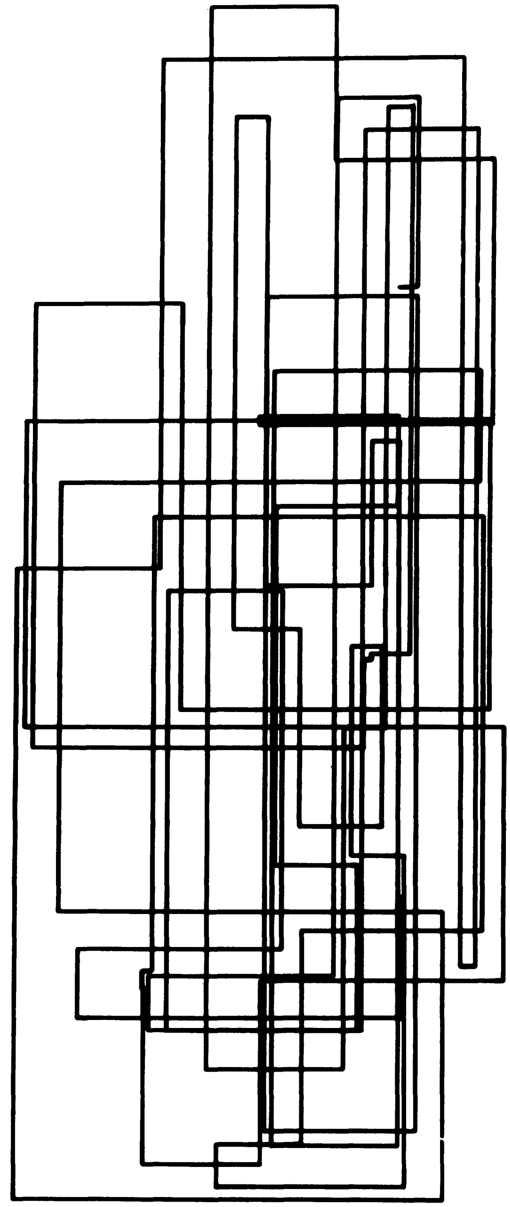
Computer-produced pictures by A. Michael Noll.

Right: "Vertical-Horizontal No. 1"

Bottom right: "Vertical-Horizontal No. 2"







Far right: "Vertical-Horizontal No. 3" © A. Michael Noll, 1965. In these pictures only one of the two coordinate values (X or Y) was changed from one point to the next along a continuous line. The change alternated between the X and the Y value. Otherwise the positions of the points were chosen at random with a uniform probability density.

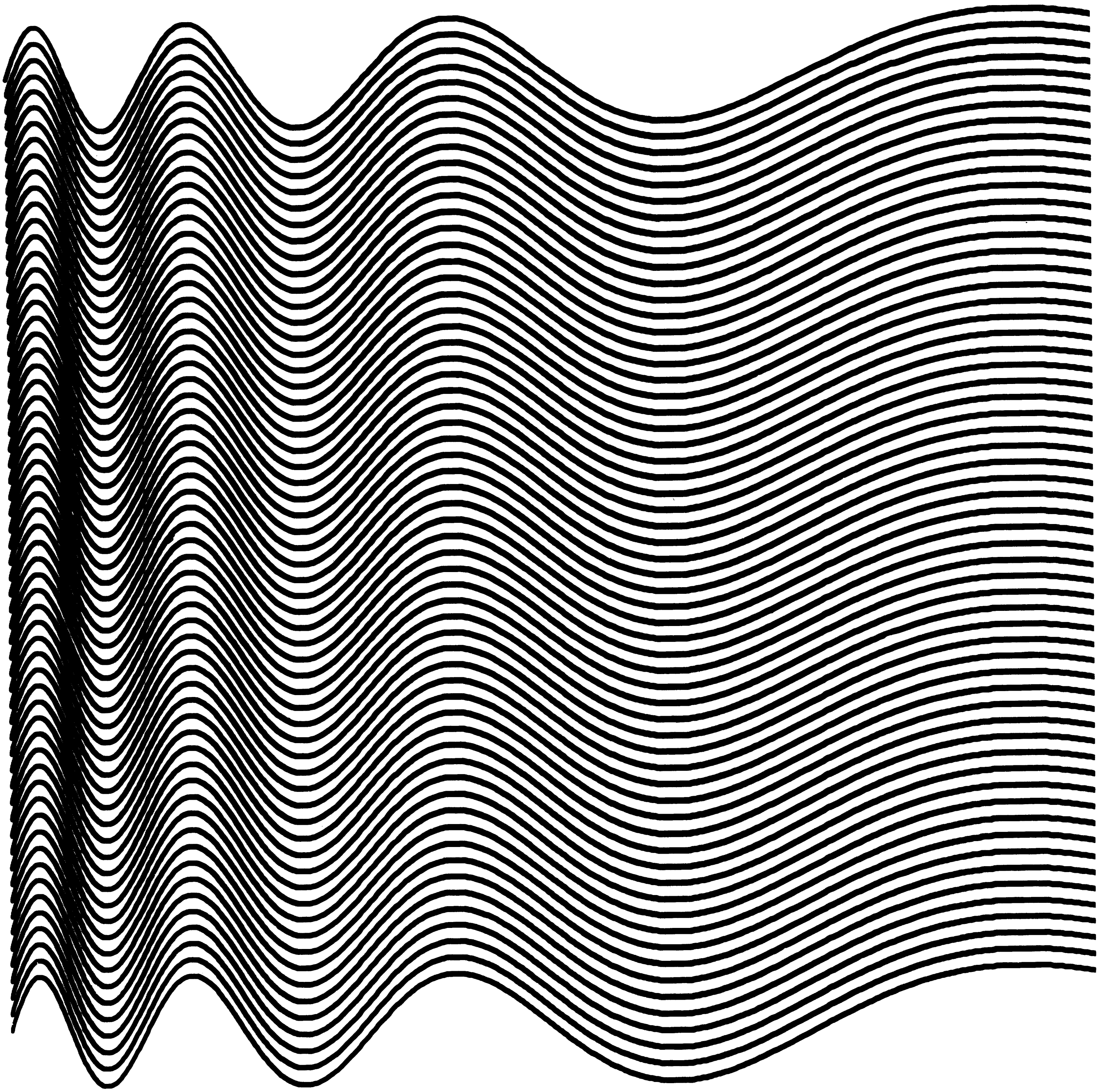
random and quasi-random abstract images. The artistic merit of the image thus generated is a matter of personal opinion, but it should be taken into account that the medium was in this case strongly limited by the exclusive application of black straight lines.

Many "op art" paintings are very regular and mathematical in design. The computer is extremely adept at constructing purely mathematical pictures and hence should be of considerable value to "op" artists. The drudgery of drawing or painting complex designs such as those in moiré patterns can be easily done by the machine. As an example, Bridget Riley's painting "Current" is a series of parallel lines that mathematically can be specified as sine waves with linearly increasing period. Such a formulation of her painting enables the computer to calculate an array of points based upon a simple mathematical formula. The microfilm plotter then connects the points to produce the finished result.

The fact that an "op art" picture can be adequately produced by a computer should not detract from the artistic merit of either the artist or the picture resulting from the artist-computer collaboration. The creative process takes place in the mind of the artist; the final painting is only the artist's rendition of his mental image. This is particularly true of "op art" although other types of art do involve the artist's interaction with his materials as part of the creative process.

Since most "op art" is definitely mathematical it is not at all surprising that the computer can duplicate "op" paintings. However, what could the computer do with different forms of abstract paintings?

This problem was approached by trying a quasi-random duplication with the computer of an abstract painting. The painting chosen for this investigation was Piet Mondrian's "Composition With Lines" (1917). This black and white



"Ninety computer-generated sinusoids with linearly increasing period." The top line of this picture was mathematically expressed as a sinusoid curve. The computer was then instructed to repeat the line 90 times. The result approximates closely Bridget Riley's painting "Current."

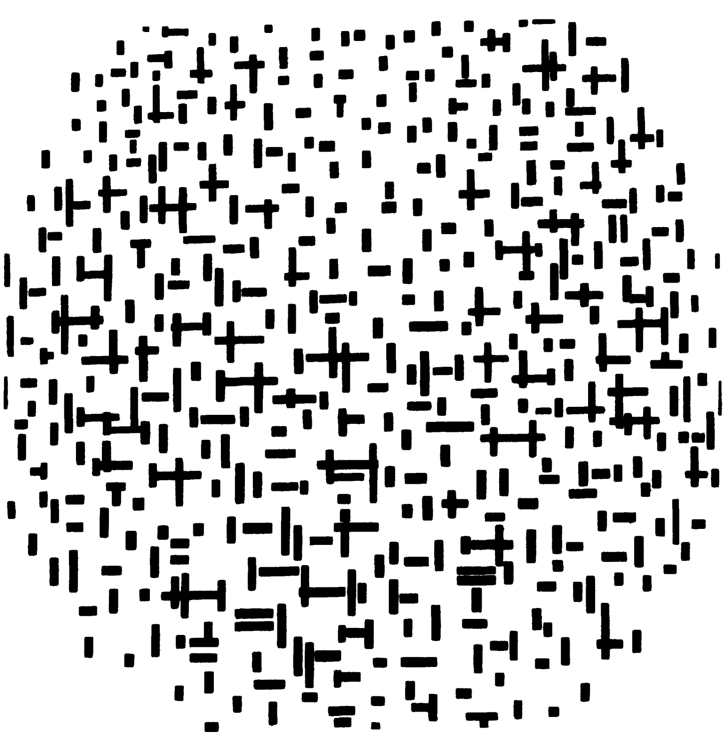
painting is from Mondrian's earlier period when he was experimenting with representations of the vertical and horizontal motifs of nature. "Composition With Lines" was chosen because it was composed entirely of solid black bars which could be drawn very easily by the microfilm plotter.

A cursory examination of Mondrian's painting reveals that, first, the outline of the painting is nearly circular except for cropping of the sides, top and bottom; second, the bars falling within a region in the upper half of the painting are shortened in length; third, the length and width of the bars otherwise appear random; and fourth, the placement of the bars is not random but seems to follow some scheme so that the entire space is almost uniformly covered.

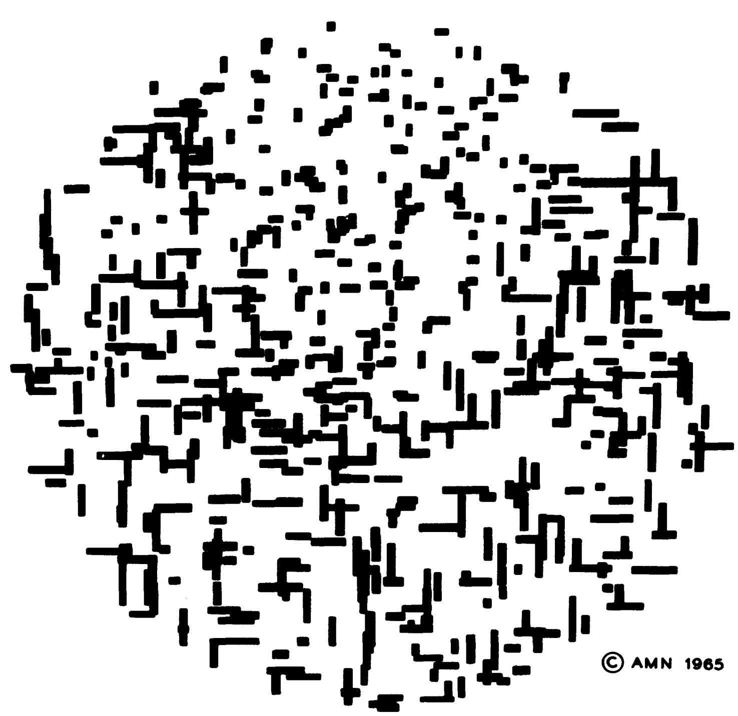
A computer program was written utilizing the first three of the observations about Mondrian's painting. The placement of the bars was random with a uniform density. The vertical and horizontal bars were approximated as a series of closely spaced, and therefore overlapping, parallel line segments. These bars were placed randomly within a circle of radius 450 units. The choice of vertical or horizontal bars was equally likely. The width of the bars was equally probable between 7 and 10 units while the length of the bars was equally probable between 10 and 60 units. If the center of the bar fell inside a certain parabolic region with its origin at the center of the circle, the length of the bar was reduced by a factor proportional to the distance of the bar from the edge of the parabola. A trial-and-error approach ensured that the final picture was reasonably similar to the Mondrian painting.

In a psychological experiment, a sample of one hundred artistically naive subjects were given reproductions of the Mondrian and of the computer picture.[1] These people were instructed to indicate which of the pictures they preferred and to identify the computer picture. In this particular group, 59% of the subjects preferred the computer picture and 28% identified correctly the computer pictures, i.e. 72% thought the Mondrian picture was done by computer. The sample group was composed of technical and clerical personnel who probably had no bias against computers. As indicated by questionary comments of these subjects, there was a trend among the non-technical people to associate randomness with creativity. Therefore, the results of this experiment neither discredit Mondrian nor imply that the computer is a greater artist than Mondrian, but raise the question to what extent randomness has aesthetical and emotional appeal. To answer this question, more experiments with controlled groups of subjects would be required.

The human-or-machine experiment, testing the preference of the two pictures, one computer-generated, the other by Mondrian, could raise a few more questions. For instance, although only one particular sequence of random numbers was used by the computer in producing this picture, it is not known if other pictures generated by statistically identical random sequences would be preferred over the Mondrian. The computer picture would clearly be more random than the Mondrian painting. However, more elaborate schemes could be used to produce pictures that even more closely resemble the Mondrian.

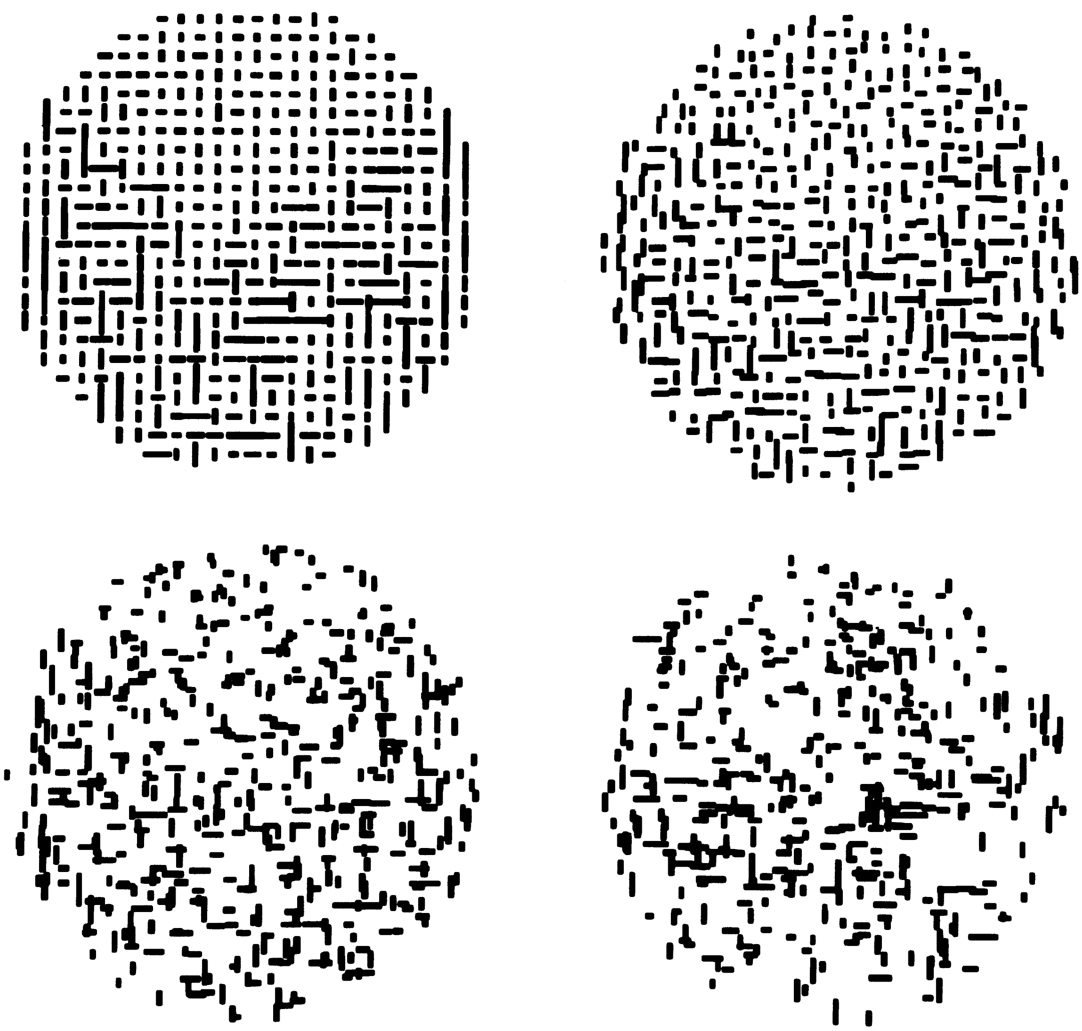


Bottom left: "Composition with Lines," (1917) by Piet Mondrian, © Rijksmuseum Kroller-Muller.



Bottom right: "Computer Composition With Lines," by A. Michael Noll in association with an IBM 7094 digital computer and General Dynamics SC-4020 microfilm plotter, © A. Michael Noll, 1965. This composition approximates Piet Mondrian's "Composition with Lines" (1917).

Four computer-generated random patterns based on the composition criteria of Mondrian's "Composition with Lines."



As a follow-up, another series of Mondrian-like computer pictures was generated. The scheme used to produce these pictures utilized random bar lengths and random bar widths within specified ranges. The bars were shortened if they fell within a parabolic region in the upper half of the picture. Only vertical bars were permitted along the sides of the picture. The actual positions of the bars were determined by adding a uniform-density random perturbation to an otherwise completely uniform grid-like set of positions. This random perturbation has a specified range; the range is zero and increases geometrically to a range of ± 250.

The conclusion of these investigations of computer-generated two-dimensional pictures is that the exciting potential of computers in art consists in their capability of producing mixtures of random elements with mathematically specified formulae for order. The experiments reported here involved only black and white pictures, but in the very near future color picture tubes will be controlled by computers, and infinitely variable color mixing will be possible. Presently, any artist desiring to use the computer would require a fairly sophisticated knowledge of computer programming. However, special "programming languages" that closely suit the needs of any particular artist could be developed, and these languages could be as natural to use as the conventional brushes and oils. Until recently the time lag between the running of the computer program and the finished picture has been several hours. Now, however, new display devices which immediately create an image on a large picture tube are being made available. Special lightpens are also available for writing and drawing on the face of the cathode ray tube and in this way it is possible to modify the picture. Such devices allow the computer-artist to sit at the console of the machine and to interact with the machine to produce a picture immediately. In the future it may even be possible for an artist to rent a console with a display device and work with a computer over distances. Many people would share the same mammoth facilities of a central utility computer. As leisure time increases and computer costs decrease, the use of time-shared computers for creative activities may become quite commonplace.

REFERENCE

1. "Human or Machine: A Subjective Comparison of Piet Mondrian's 'Composition With Lines' and a Computer-Generated Picture," *The Psychological Record,* Vol. 16, No. 1, January 1966, pp. 1-10.

ACKNOWLEDGEMENTS

We gratefully acknowledge the assistance of the following individuals, firms and corporations in the preparation of material for this issue:

Dr. Martin Krampen, Institute of Design, University of Waterloo, Ontario, Canada and Hochschule für Gestaltung, Ulm, Germany;

PLANNING AND DESIGN 66, a conference on computers in design and communication, June 1966, at the University of Waterloo, Ontario, Canada. The conference was arranged by Prof. Martin Krampen and its speakers contributed most of the articles presented in this issue;

Prof. Allen Bernholtz, Harvard University, Cambridge, Massachusetts and Edward Bierstone, University of Toronto, Toronto, Canada;

Prof. Steven A. Coons, Massachusetts Institute of Technology, Cambridge, Massachusetts;

William A. Fetter, Boeing Co., Seattle, Washington;

Edwin L. Jacks, General Motors Research Laboratories, Warren, Michigan. The article by Mr. Jacks first appeared as "DAC-1: A Computer System for the Study of Graphical Man-Machine Communication" in the General Motors Engineering Journal, Vol. 12, No. 2, and is published here by permission of General Motors Corporation;

Dr. Kenneth C. Knowlton, Bell Telephone Laboratories, Murray Hill, New Jersey;

Dr. Marvin L. Manheim, Massachusetts Institute of Technology, Cambridge, Massachusetts;

A. Michael Noll, Bell Telephone Laboratories, Murray Hill, New Jersey;

Kenneth G. Scheid, K. G. Scheid & Associates, Pittsburgh, Pennsylvania.

Arthur E. Neuman, Control Data, Minneapolis, Minnesota.

PHOTOGRAPHIC CREDITS

Christopher Alexander and Marvin L. Manheim, M.I.T., Cambridge, Mass.: 35, 36, 38

Allen Bernholtz, Harvard University, Cambridge, Mass. and Edward Bierstone, Toronto Canada: 40, 41, 42, 47, 48, 49, 50, 51

Dr. Carol Bosche, Bell Telephone Laboratories, Murray Hill, N. J.: 62

Boston Architectural Center, Boston, Mass.: 10

Digital Equipment Corp., Maynard, Mass.: 55

William A. Fetter, Boeing Co., Seattle, Wash.: 14, 16, 17, 18, 19, 20, 21, 22, 23

Barrett Hargreaves, General Motors Research Laboratories, Warren, Mich.: 24, 26, 27, 28

International Business Machines Corp., Chicago, Ill.: 54

Edwin L. Jacks, General Motors Research Laboratories, Warren, Mich.: 24, 26, 27, 28

Dr. Bela Julesz, Bell Telephone Laboratories, Murray Hill, N. J.: 62

Kenneth C. Knowlton, Bell Telephone Laboratories, Murray Hill, N. J.: 58, 60, 61, 63

Lockheed-Georgia Co., Marietta, Ga.: 12

Marvin L. Manheim, M.I.T., Cambridge, Mass.: 30, 32, 33, 37

Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, Mass.: 8, 9, 11, 30, 32

Mergenthaler Linotype Co., Brooklyn, N. Y.: 56

A. Michael Noll, Bell Telephone Laboratories, Murray Hill, N. J.: 64, 66, 67, 68, 69, 70, 71

Photon, Inc., Wilmington, Mass:. 52

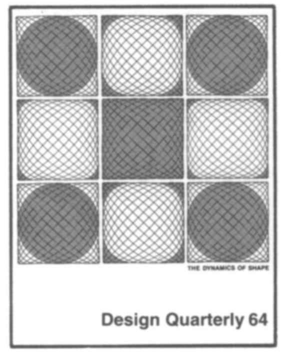
Larry G. Roberts, M.I.T., Lincoln Laboratory, Lexington, Mass.: 6, 13

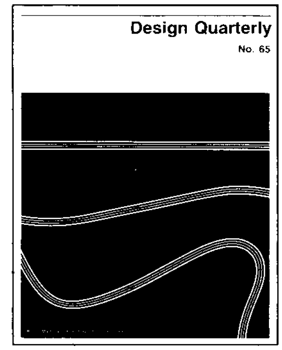
Recognition Equipment Inc., Dallas, Texas: 54

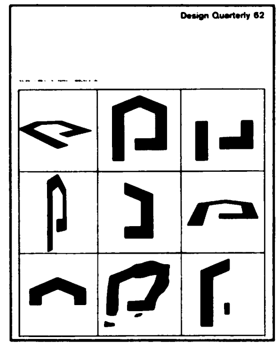
Rijksmuseum Kröller-Müller, Otterlo, The Netherlands: 70

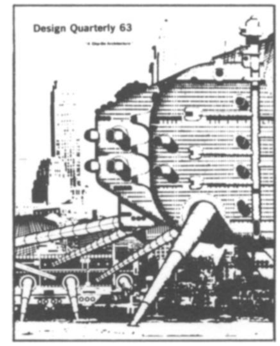
DESIGN QUARTERLY issues available: single issues 60c each, double issues (numbers 18-19, 48-49, 50, 51-52) $1.60 each. Beginning with 61, single issues $1.00, double issues $2.00.

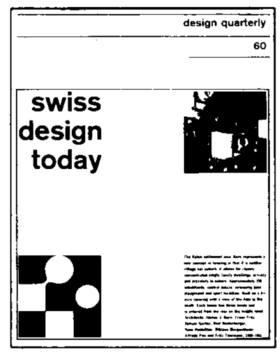
Number 9 Outdoor Furniture and Accessories
10 Product Review 1949
11 Textiles and Designers
12 Lamps and Lighting
13 Everyday Art Exhibitions
14 Useful Objects / Work of Alvin Lustig
15 Tradition in Good Design
16 Tradition in Good Design: 1940-1950
17 Product Review 1951
18-19 Knife/Fork/Spoon
20 Contemporary Chairs
21 Product Review 1952
23 Made in Sweden
24 Product Review 1953
25 Fabric Designers and Their Work
26 Product Review: 20th Century Ballet Design
27 Five Ceramists and Their Work
28 Furniture Designers and Their Work
29 American design
30 Critical articles on contemporary painting and architecture
31 Contemporary book design
32 Tenth Triennale Product Review
33 84 Contemporary Jewelers
34 The story of Orrefors glass
35 Product Review 1956
36 Eight British designers and their work
37 Contemporary Finnish designers and their work
38 Product Review 1957
39 Eight designer-craftsmen
40 Industrial design in Germany
41 Product Review 1958
47 Product Review 1960
48-49 Fifty-seven American weavers and their work—
50 Art and Design in Hawaii—double issue
51-52 Japan: Design Today
53 Marcel Breuer
54 MacKenzie Pottery: Mendota Sculpture Foundry
55 Japan: Book Design
57 Children's Furniture
58 Tyrone Guthrie Theatre
59 Industrial Design in the Netherlands
60 Swiss Design Today
61 The 13th Triennale
62 Signs and Symbols in Graphic Communication
63 A Clip-On Architecture
65 Bruno Mathsson: Furniture/Structures/Ideas
66-67 Design and the Computer

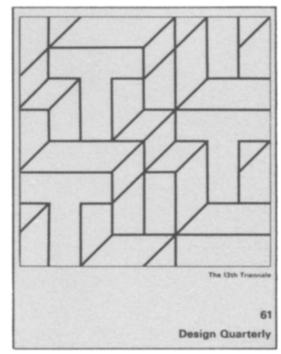


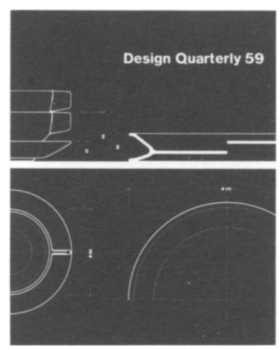


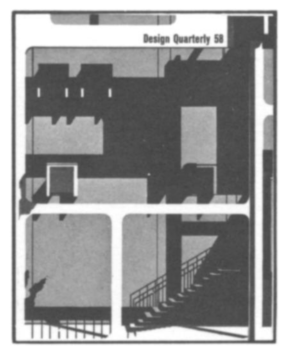












DE SIG N
A N D THE
C OMP UT ER
D E
S I
G N
A ND T HE
COM P U TE R

Walker Art Center / 1710 Lyndale Avenue South / Minneapolis, Minnesota 55403