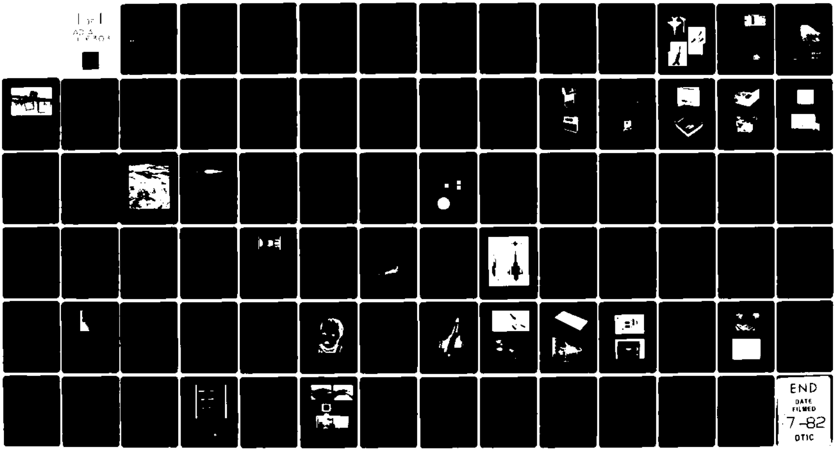
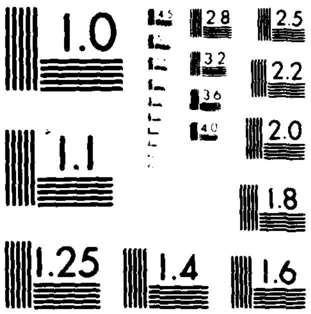
AD-A116 303 AIR FORCE WEAPONS LAB KIRTLAND AFB NM F/G 9/2
COMPUTER GRAPHICS FOR SCIENTIFIC APPLICATIONS.(U)
FEB 82 L A FELDMAN, N L RAPAGNANI
UNCLASSIFIED AFWL-TR-80-138 NL



END
DATE
FILMED
7-82
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU

AFWL-TR-80-138

2

AFWL-TR-
80-138

AD A116303

COMPUTER GRAPHICS FOR SCIENTIFIC APPLICATIONS

Laurence A. Feldman
Nazareno L. Rapagnani

February 1982

Final Report

Approved for public release; distribution unlimited.

JUN 28 1982

A

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

82 06 28 065

AFWL-TR-80-138

This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, under Job Order 33260690. Mr. Larry Rapagnani (ARAC) was the Laboratory Project Officer-in-Charge.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been authored by an employee of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the material contained herein, or allow others to do so, for the United States Government purposes.

The Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

If your address has changed, if you wish to be removed from our mailing list, or if your organization no longer employs the addressee, please notify AFWL/ARAC, Kirtland AFB, NM 87117 to help us maintain a current mailing list.

This technical report has been reviewed and is approved for publication.

Larry Rapagnani

LARRY RAPAGNANI
Project Officer

FOR THE COMMANDER

David W. Seegmiller

ROBERT F. BESTGEN
Lt Colonel, USAF
Chief, Chemical Laser Branch

DAVID W. SEEGMILLER
Colonel, USAF
Chief, Advanced Laser Technology Div

DO NOT RETURN COPIES OF THIS REPORT UNLESS CONTRACTUAL OBLIGATIONS OR NOTICE ON A SPECIFIC DOCUMENT REQUIRES THAT IT BE RETURNED.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWL-TR-80-138	2. GOVT ACCESSION NO. AD-A116 303	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER GRAPHICS FOR SCIENTIFIC APPLICATIONS		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Laurence A. Feldman Nazareno L. Rapagnani		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Weapons Laboratory (ARAC) Kirtland Air Force Base, NM 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F/33260690
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (ARAC) Kirtland Air Force Base, NM 87117		12. REPORT DATE February 1982
		13. NUMBER OF PAGES 78
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Graphics	Vector
Finite Element	Grey-Level
Finite Difference	Color
Raster	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents an overview of the subject of computer graphics and its potential contribution in the analysis of scientific/engineering data, ~~and~~ is directed to the engineer familiar with computer modeling and analysis but not with computer visualization techniques important in treating complex results. Standard graphics hardware and software are discussed, with emphasis on recent advances made by the computer graphics industry.

(over)

DD FORM 1 JAN 73 1473

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

Covered are diverse technical applications including two-/and three-dimensional dynamic simulations and the software techniques employed to "visualize" the results of these calculations. Realism--employing shading algorithms to make artificial graphics appear real and lifelike--is investigated.

An approach to writing a complete multidimensional scientific animation graphics package--SCAN--is detailed. The challenge of dealing with the voluminous data generated by three-dimensional numerical modeling encouraged the development of such a program. The objectives, architecture, and results are briefly described; and future trends in computer graphics and their immediate impact on existing computer analysis, ~~is~~ explored.

A

CONTENTS

Section		Page
I	INTRODUCTION	5
II	COMPUTER GRAPHICS	10
	1. SELECTING A HOST COMPUTER FOR COMPUTER GRAPHICS	10
	2. SELECTING A GRAPHICS DEVICE	12
	3. SOFTWARE TECHNIQUES	17
III	CATEGORIZING ENGINEERING DATA	24
	1. DIMENSIONALITY	26
	2. NUMERICAL SIMULATION	27
IV	SOFTWARE OBJECTIVES	32
V	BUILDING BLOCKS OF A GRAPHICS SOFTWARE PACKAGE	39
	1. GRAPHICS TOOLS FOR LINE PLOTTING	39
	a. Contouring	39
	b. Topography	42
	c. Modeling a 3-D scene	42
	d. Vectors	48
	e. Tracer particles	48
	2. GRAPHICS TOOLS FOR SURFACE PLOTTING (SHADE, COLOR)	48
	a. Visible surface problem	52
	b. Shading	53
	c. Realism	62
VI	ANIMATION AND REAL TIME	67
VII	FUTURE TRENDS IN COMPUTER GRAPHICS	71
	REFERENCES	73
	GLOSSARY	74

ILLUSTRATIONS

Figure		Page
1	Computer graphics. Scientific graphs generated by 'SCAN', a FORTRAN computer graphics program	6
2	Advertisements. Computer graphics generated by Evans & Sutherland, Salt Lake City, Utah	9
3	Graphics devices	18
4	Surface of Mars. Employing a 'distribution half-tone' algorithm on a Benson-Varian electrostatic plotter. Data from Mariner exploration. Hans Morvec, Artificial Intelligence Laboratory, Stanford University, Calif.	25
5	Aerodynamic body. Pressure contours of 2-D transonic flow about an M-117 weapon system	26
6	Eulerian/Lagrangian coordinate system	30
7	Lagrangian deformation. A detonation wave propagates outward causing deformation of the casing	31
8	Device independence. "SCAN," a complete scientific graphics package is written in standard FORTRAN with plot emulators creating "portability" and "device independence"	33
9	Contouring. A simple algorithm for 2-D plotting	40
10	Shock in a nozzle. Pressure contours in a nozzle. Coalescing contours indicate a shock	41
11	Turbulent flow in 3-D. Orthogonal contours of turbulent kinetic energy	41
12	Energy waves. Pressure of a 2-D wave oscillating in a square; 3-D surface mapping employed	42
13	Space shuttle. USS Enterprise space shuttle courtesy of M & S Computing	43
14	F-16 aircraft. Employing one data base and simply 'viewing' from three different perspectives, the top, front, and side views are produced	45
15	Fuel tank assembly. Generating a fuel tank for an F-16 by employing parallelpipeds	46
16	Electric current density	49

ILLUSTRATIONS (Continued)

Figure		Page
17	Vectors in 3-D. *Vectors in three dimensions are* plotted with components attached to the tail to clarify direction	50
18	Vorticities in turbulent flow	51
19	Tracer particles in a combustion chamber. Following the flow in a chamber by 'observing' the path of tracer particles	52
20	Scott at 2	55
21	Raster scan of boy's face	56
22	Oblique view of F-16	58
23	Laser device. Employing color to emphasize pressure regions in a contour, 3-D surface map	59
24	Subsonic flow. Employing color to highlight subsonic region in wind tunnel flow	59
25	Vectors in color	60
26	Turbulent 3-D tracer particles	60
27	MHD generator	61
28	Realism. Curved surfaces, shadows, transparency, reflection, hidden surface. Turner Whitted, Bell Laboratories, Holmdel, N.J.	63
29	DNA molecules. Nelson Max, Lawrence Livermore Laboratory, Livermore, Ca	63
30	Plasma flow. Magnetohydrodynamic coupling between gas dynamic and electromagnetic force fields - in animation	68
31	F-16 in a roll. Animation	68
32	Real time. Simulation of an aircraft landing on a runway. Evans & Sutherland, Salt Lake City, Utah	70

TABLES

Table		Page
1	Evaluation of graphics devices	13
2	Vector and raster scan graphs	35
3	Fuel tank coordinates	47
4	Digital shading	54

I. INTRODUCTION*

Effective modeling of physical phenomena on the computer involves an understanding of five critical areas: computer hardware, physics, numerics, software development, and graphics display. This report briefly discusses the first four topics but concentrates primarily on the last, computer graphics.

Vast new computer graphics techniques exist today (Ref. 1) which can aid immeasurably in the analysis of complex scientific and engineering data. The feasibility of sophisticated graphics/image generation for creating computer produced "realism" offers the theorist enhanced opportunities to comprehend the full impact of his calculation and, equally as important, to facilitate its communication to others less intimate with its development.

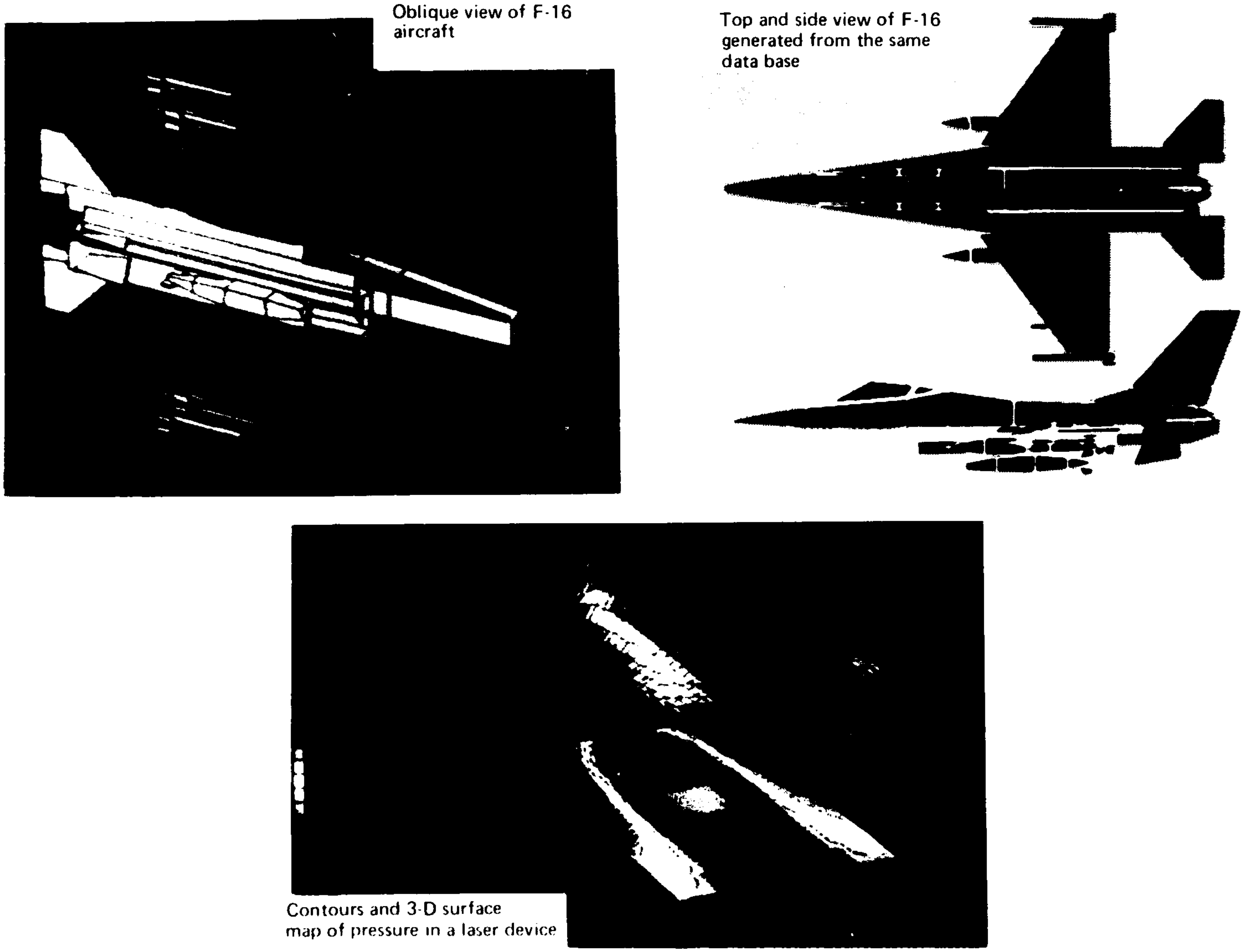
The subject of this paper is computer graphics and its application to a diversity of problems encompassing all of the engineering disciplines. A discussion of the use of computer hardware and software in generating visual output is presented, with particular interest focused on recent advances made by the computer industry in facilitating exploitation of computer visualization

As a brief introduction into the diverse types of computer graphics available to the engineer, consider the 3-D perspectives, color, half-tone, raster scan, etc., illustrated in Figure 1. SCAN, a software package which generated these pictures, will be detailed throughout this report. In addition to these plots, consider the excellent computer graphics produced for the advertising industry (Fig. 2) by employing computer hardware and graphics software techniques well within the reach of the engineering community.

---

1. Newman, W. H. and Sproull, R. F., Principles of Interactive Computer Graphics, 2nd Ed., McGraw Hill, 1979.

*The Glossary lists some of the more common terms used in the specialized computer graphics vocabulary.

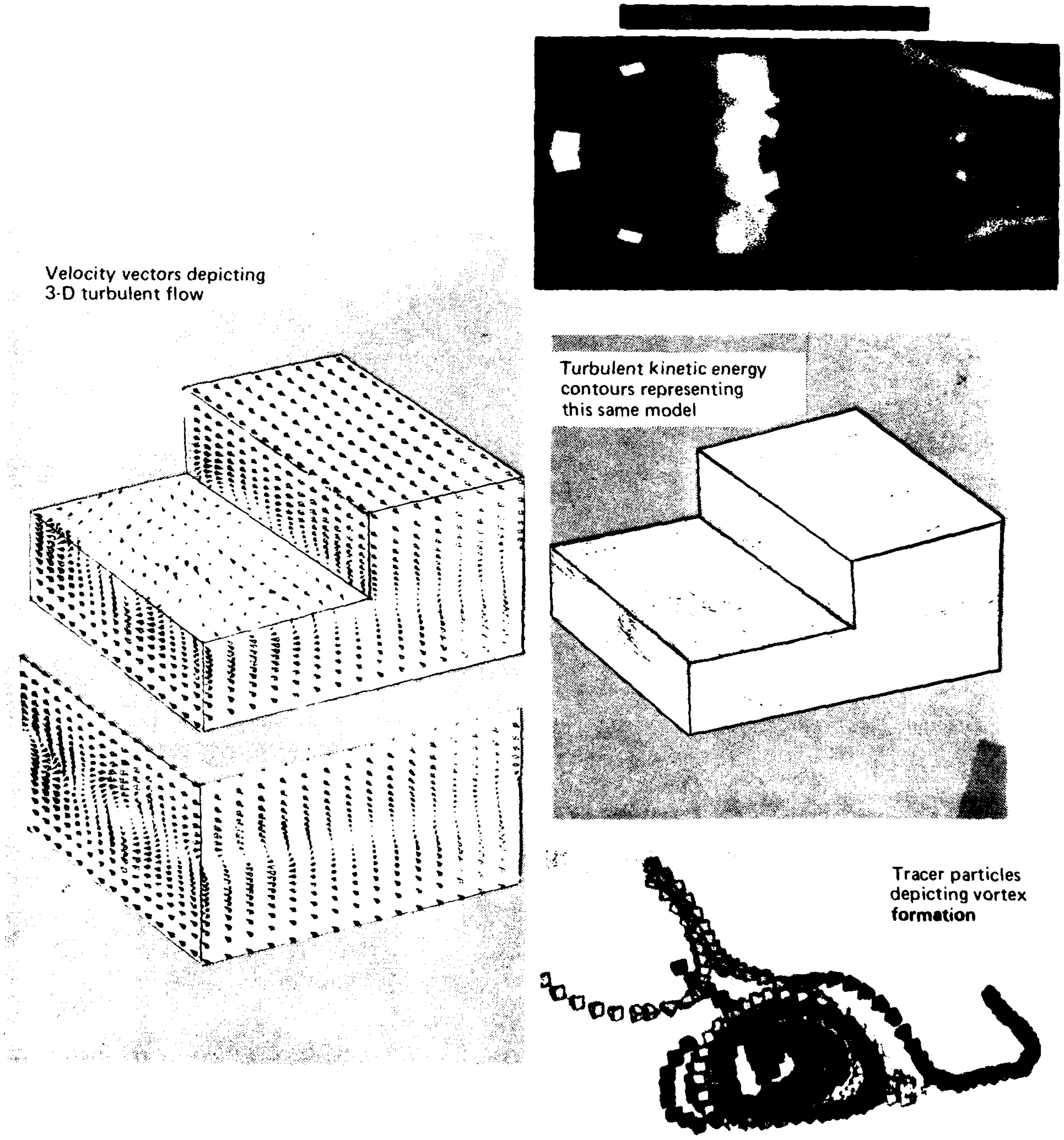


Oblique view of F-16 aircraft

Top and side view of F-16 generated from the same data base

Contours and 3-D surface map of pressure in a laser device

Figure 1. Computer graphics. Scientific graphs generated by 'SCAN', a FORTRAN computer graphics program.



Temperature distribution in a laser device

Velocity vectors depicting 3-D turbulent flow

Turbulent kinetic energy contours representing this same model

Tracer particles depicting vortex **formation**

Figure 1. (Continued).

Baby's face on an electrostatic plotter

Subsonic flow in a wind tunnel, supersonic 'shading' suppressed

Baby's face on an impact printer

Schematic of energy conversion device. Grey scale indicates plasma flow characteristics

Baby's face on an image processor

Figure 1. (Concluded).

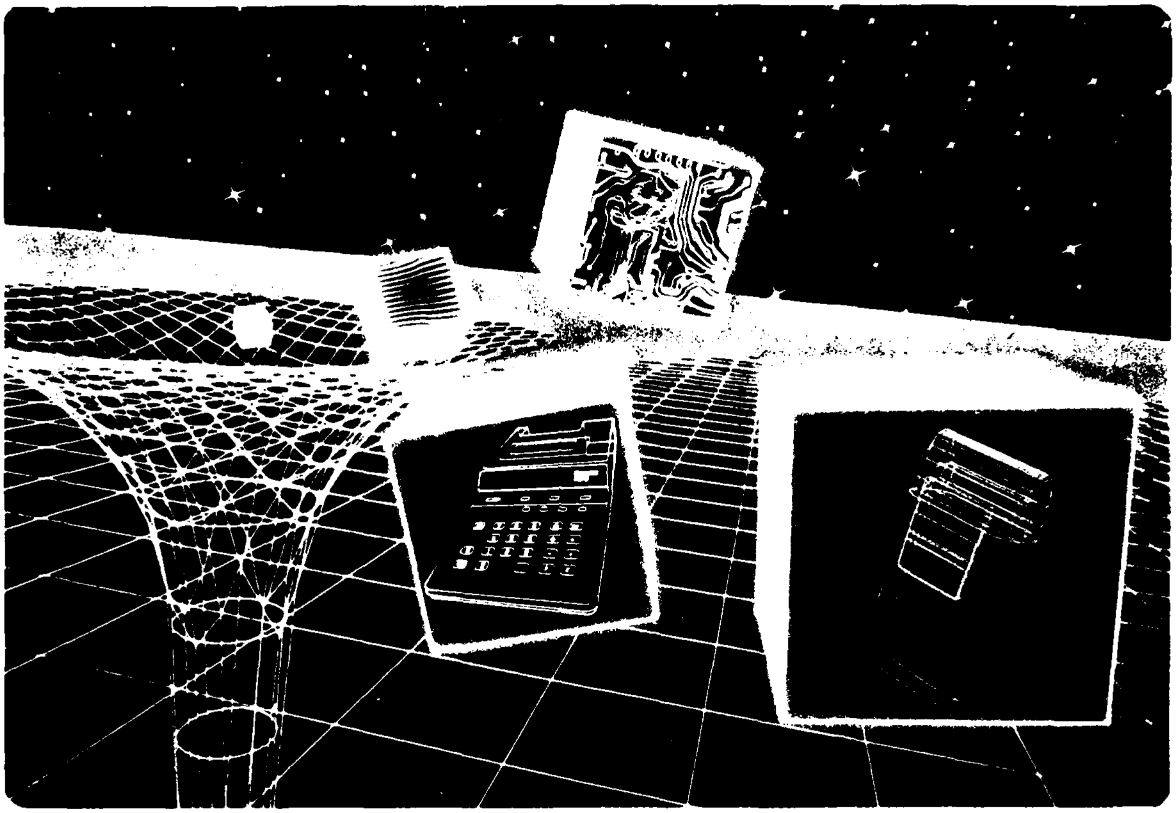


Figure 2. Advertisements. Computer graphics generated by Evans & Sutherland, Salt Lake City, Utah.

The same technology employed by the advertising industry* to generate sophisticated computer graphics (Fig. 2) is being exploited by the engineering community to produce technical graphs of equal quality.

---

*Evans and Sutherland.

II. COMPUTER GRAPHICS

Computer graphics, "the creation and manipulation of pictures with the aid of computers" (Ref. 1), has seen rapid growth in the late seventies for a number of reasons, not the least of which has been the development of high speed "number crunchers," reasonably fast minicomputers with unlimited (virtual) memory and high-resolution, inexpensive storage tube displays. The computer industry as accelerated the exploitation of graphics technology because these major advances do not substantially increase product cost. Concentrating on these three critical aspects: speed, memory and resolution, a discussion of establishing a complete computer graphics system to treat diverse engineering applications follows.

There are three stages in the generation of scientific graphs: (1) modeling the physical application on the host computer, (2) converting the results of this model to graphics commands (lines, polygons, color, etc.), and (3) transmitting these graphic commands to the graphics device which generates the picture output. An investigation of hardware requirements (i.e., host computer and graphics device) follows, with attention given to engineering modeling and graphics software requirements later in the section.

1. SELECTING A HOST COMPUTER FOR COMPUTER GRAPHICS

The selection of a host computer is usually a moot issue, since most computer facilities are already "entrenched" in a designated computer system for about 5 or 10 years. Aside from the principal functions of the computer--to perform a scientific calculation and generate output to be analyzed--certain desirable computer features relating to graphics become important, namely speed and memory. Central Processor Unit (CPU) speed is important because of the great quantity of data that has to be treated and because graphics software execution can be time consuming, especially when employing sophisticated algorithms to be discussed later. Input/Output (I/O) time can also be significant, depending on how the transfer of data is accomplished. A practical rule of thumb is that, for every dollar spent on central, peripheral and I/O processing, it is not unreasonable to spend an additional 10 to 15 cents for graphics, especially since without graphics, most of the information generated by the calculation goes unrecognized.

Memory is always a problem with any "discretization" model, especially when treating grids of many thousands of cells, but it is just as critical in computer graphics because of the great quantity of data to be sorted, searched, manipulated, etc.; and if all the data cannot reside in the core, then frequent "buffering" to the disk can create an I/O problem which will slow the generation of the picture significantly.

It is easy to grasp the importance of having sufficient memory on the host computer by considering the manner in which pictures are displayed on the raster scan device, e.g., Cathode-Ray Tube (CRT). A television picture has approximately 500 dots (nibs, pixels, points, etc.) in both the horizontal and vertical directions. One computer-driven picture will require 250,000 (500 x 500) addressable picture elements just to "map" an intensity value to each dot on the screen. Since all this memory is not usually readily available, the only alternative is to perform horizontal (raster) scans one line at a time and retain the remaining data on the disk. Once again, the I/O problem occurs.

Having adequate memory--real or virtual--alleviates many of the graphics logistic problems and avoids unnecessary calculation and bookkeeping when dividing a picture into smaller segments in order not to exceed limited memory allocations. Since modestly priced minicomputers (e.g., the DEC's VAX) are now available that operate at speeds comparable with mainframes and have virtual memory, they represent an alternative to large mainframes for accomplishing scientific simulation and graphics.

A severe problem with large mainframes is that they become overloaded with numerous time-share users, and generating complicated images on a plot device can become a lengthy procedure because of slow communication between the host and the graphics device. As an example, listing graphics commands from the host to a color graphics terminal may take minutes under normal circumstances; if, however, the system is overworked, picture generation could take an hour or more. "Interactive graphics"--communicating back and forth between the computer and graphics device--is also tedious when the host computer is lethargic. A dedicated minicomputer, however, usually avoids these problems, since the computer is free to "concentrate" on a few tasks as opposed to a hundred or more found on large time-sharing mainframes.

In conclusion, the best host computer is one with adequate memory (> 512K), sufficient speed, and few users. In support of this philosophy, small computer systems are being designed as "dedicated graphics machines,"* devoting all hardware/software resources to the sole objective of achieving "high-performance graphics."

2. SELECTING A GRAPHICS DEVICE

A graphics device is a computer hardware instrument whose sole function is to display pictures from instructions received from a computer. Computer and graphics devices are now separate entities, but are linked either "on-line" (the computer sends graphics commands as soon as they are generated) or "off-line" (the computer stores the data on perhaps a tape) where the plot data are processed by the graphics device later.

The device may be as primitive as a line printer, or increase in sophistication to a line plotter, an electrostatic plotter, a raster graphics terminal, the ultimate in high-resolution image processors, or film recorders.

Selecting a graphics device is based on the following considerations:

a. Hardware costs
b. Output (paper, film, polaroid) costs
c. Host computer execution time
d. Plot turn-around time
e. Processing difficulties
f. Flexibility in treating diverse applications
g. Color/grey-level
h. Picture resolution

In general the selection of the optimum graphics device is dictated by the desired quality and degree of sophistication of the output. Table 1 lists the merits and shortcomings of six types of available plotters.

Graphics devices can be categorized into two groups: (1) line or stroke plotters and (2) surface plotters or raster scan. Line plotters connect points on a graph either mechanically--moving a pen from point to point--or on a CRT, generating an electron beam which scans across the screen starting at

---

*Superset's Personal Graphics Machine (PGM) is a 48-bit, virtual memory stand-alone minicomputer system capable of driving extended FORTRAN graphics programs of over 40,000 lines at speeds and accuracy competitive with mid-sized computers.

TABLE 1. EVALUATION OF GRAPHICS DEVICES

DEVICE	IMPACT PRINTERS	PEN PLOTTERS	ELECTROSTATIC PLOTTER	RASTER GRAPHICS TERMINAL	IMAGE PROCESSOR	FILM RECORDER
MANUFAC-TURER	Printronix	Calcomp Soltec Varian	Benson-Varian Versatek	Hewlett Packard Tektronix Chromatics Ramtek	Grinell Genisco De Anza Systems Evans Sutherland Ikonas Lexidata Megatek	Dicomed Information International Inc. (III)
APPROXI-IMATE COST	$700 to $6,000	$700 to $30,000	$6,000 to $15,000	$5,000 to $20,000	$20,000 to $80,000	approx $300,000
OUTPUT	paper	paper	paper	CRT	CRT	film
TURN-AROUND TIME	seconds	minutes to hours	seconds	seconds to mins.	seconds	days
PROCESSING	fast	slow	fast	very fast	very fast	very slow
DIFFI-CULTIES	none	usually off-line	none	none	none	film must be processed
HOST COMPUTER TIME	Negligible	Negligible	low	low	can be significant for the types of plots one would generate	

TABLE 1. *(Concluded)*

DEVICE	IMPACT PRINTERS	PEN PLOTTERS	ELECTROSTATIC PLOTTER	RASTER GRAPHICS TERMINAL	IMAGE PROCESSOR	FILM RECORDER
FLEXIBILITY	limited	only simple line plots	very flexible	realism flexible	realism flexible	realism flexible
COLOR	yes	1-8	no	8 or more	up to 4024 simult.	essent. infinite
GREY-LEVEL	using half-tone and overstrike	no	using half tone	up to 8 grey-levels	very effective	effective
RESOLUTION	low (128 x 128)	high	2000 x 2000	approx. 512 x 512	512 x 512 1024 x 1024	16000 x 16000
QUALITY OF OUTPUT	fair	good	good	good	excellent	excellent
ANIMATION	no	no	no	yes	good	outstanding

one point and terminating at another. Although line plotters "draw" lines efficiently, they cannot "shade," and thus are limited to nonrealism applications.

Raster devices operate quite differently, since the picture or graph is represented by a matrix of dots with plot commands signaling those dots which are turned on (and to what color and intensity) or off (retains background color and intensity). The trend in graphics software has shifted recently toward the raster device because of the progress made by the computer industry in CRT monitors and low-cost memory development. Graphics terminals are either of the line-plotting (stroke) or surface-plotting (raster) type; but, as is the case with all image processors, the trend is toward the raster technique. Resolution is determined by the number of discernible dots per inch, with electrostatic plotters (non-CRT) offering up to 300 nibs/inch (90,000 nibs^3/in^2) and CRT systems offering 25, 50 or 100 dots/inch. Monitor resolution is usually described by the picture or screen resolution of 256 x 256, 512 x 512, 1024 x 1024 dots or some combination thereof. A television picture has a screen resolution of approximately 510 x 480.

Color or grey-level is accommodated by these systems by representing the matrix of dots on the screen by a memory (frame buffer) which stores the intensity value or color to each dot or element of the matrix. This three-dimensional memory (x,y location of pixel and intensity value) may consist of any number of bits which increases as the number of intensities (colors and grey-levels) increases. Since a binary bit is either '0' or '1,' a one-bit system (2**1) can have the beam either on or off, black or white. A two-bit system can have 2**2 intensities or 4. An eight-bit (byte) system has 2**8 or 256 simultaneous color intensities, etc. Unfortunately, computer memory requirements drive up the price of the graphics hardware at a rate proportional to the number of elements in the three-dimensional matrix (x,y, intensity) so that an eight-bit 512 x 512 picture requires 16 times the memory of a one-bit 256 x 512 system.

A raster device is capable of shading, coloring or "fill," unlike its counterpart the line plotter. Even a one-bit electrostatic plotter can employ grey-level quite effectively by appropriately selecting the proper spacing of dots to create the desired grey-level.

When discussing resolution of a picture, geometric (number of dots per inch) and color (grey-scale) resolution have to be considered simultaneously because of a 512 x 512 picture of eight colors will not appear to have nearly the same resolution as a 256 x 256 grid of 256 colors. A TV picture tube has low geometrical resolution but high color resolution, creating the impact of a sharp clear picture although a diagonal line is really a series of disjointed uneven lines (aliasing).

Also categorized under raster devices are graphics terminals and image processors. Both employ monitors and essentially operate under the same principles. Each is linked to a host computer, has a video controller to process the plot data to be sent to the monitor (including memory for the picture matrix), and a "color look-up table." The difference between the two lies in high resolution both spatially and in color for the image processor, allowing it to process the data more effectively. An image processor adds, subtracts, multiplies and divides frame buffers containing data of the same image in order to improve the quality of visual communication. Also, image processors usually employ direct memory access (DMA) to expedite the transfer of data from the computer to the monitor. Graphics terminals may in many cases transfer data over a telecommunication line, transmitting at baud rates of from 300 to 9600, which slows picture generation to minutes or even hours.

Finally, the last plot device to be discussed is a film recorder, which is an expensive camera system (16mm, 35mm) capable of reading data, usually off a tape, generating the picture internally, and shooting consecutive frames automatically. Film recorders have much higher spatial resolution (e.g., 16000 x 16000) because they are not limited to CRT monitor constraints of 1024 x 1024. Color resolution is also very high. One major disadvantage of film recorders is delay caused by film processing; whereas on a graphics terminal or image processor, the turnaround is usually within minutes.

This section would not be complete without a brief discussion of monitors or cathode-ray tubes. A monitor is a cathode-ray tube device which receives information from a microprocessor and converts it to an image on the screen by employing an electron gun (three guns for color) which emits electrons directd through a "shadow mask" to impact on a phosphorescent screen. The digital through signal from the video controller is converted to an analog signal by a digital-to-analog converter (DAC) and then transmitted to the monitor or CRT. Color is generated by assigning each of the three guns to the basic colors of red, green and blue (RGB) and creating any color by using different combinations of intensities of these three guns and by filtering RGB. The beams generated by these guns scan the picture tube horizontally at rates of 30 or 60 Hz. "Interlaced" monitors perform horizontal scans on every other line (field) producing flicker or a noninterlaced "refresh memory" rate of 60 Hz, resulting in a flicker-free picture. The memory bank storing the digital data is the frame buffer, which is a random access memory (RAM) integrated circuit and varies from one-bit (either 0 or 1) to 72 bits in depth. Usually eight bits occur where three bits may be assigned to red, three to green, and two to blue. Thus the number of possible color combinations of red, green and blue is a permutation of 2**3 x 2**3 x 2**2 = 256 colors.

Colors are defined by hue, saturation, and luminosity, where hue refers to the percentage of red, green and blue; saturation is the extent white is mixed with the hue; and luminosity is the intensity per unit area.

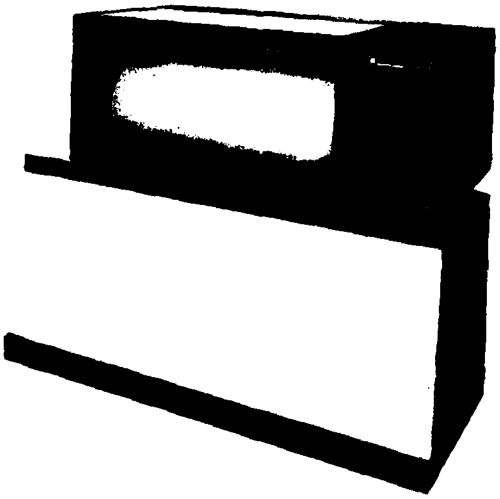
As an example of the graphics devices discussed above, Figure 3 illustrates (a) an impact printer, (b) drum plotter, (c) pen plotter, (d) raster graphics terminal, (e) image processor, (f) electrostatic plotter, (g) hardcopy device, (h) sample output, (i) landsat map generated on a raster scan device and (j) film recorder. An often overlooked, yet fundamental, problem to CRT systems is the difficulty in generating hardcopy--paper output. The most common technique is to photograph the screen with a 35mm camera, but delays are incurred due to film processing. There are devices which generate B/W and color hardcopies, but the cost of these devices are still significantly high. A somewhat lower priced grey-level hardcopy device (Tektronix) (Fig. 3g) can reproduce a sharp and realistic image (the tank) by employing only 12-16 grey levels.

3. SOFTWARE TECHNIQUES

Just as there are numerous graphics devices available, there are many techniques to display the same data. Hopefully, the fundamental question of just how to graphically represent a set of data, in order to sift through and separate the important from the irrelevant, can now be answered.

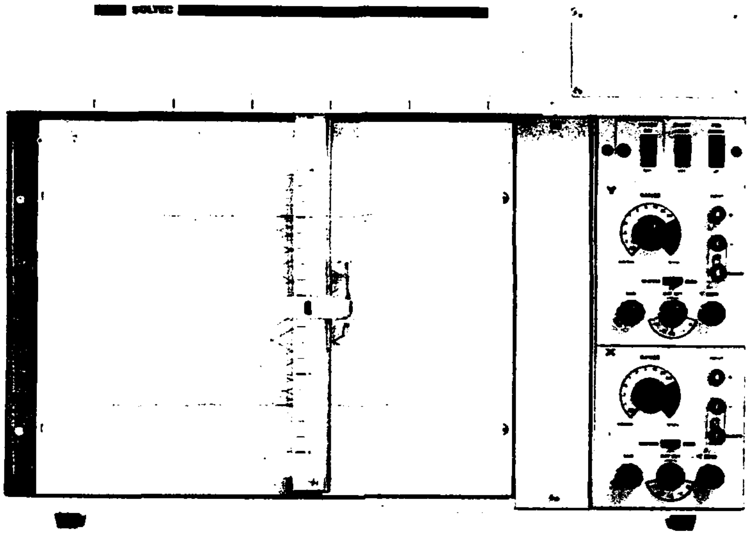
The graphics software specialist uses the following techniques to display multidimensional scientific data:

(a) Impact Printer - Printronix, Irvine, Calif.

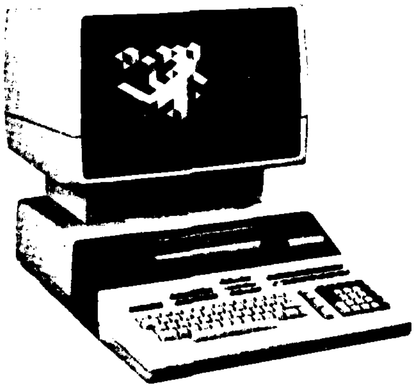


(b) Curved bed drum plotter, Benson-Varian, Mt. View, Calif.

Figure 3. Graphics devices.

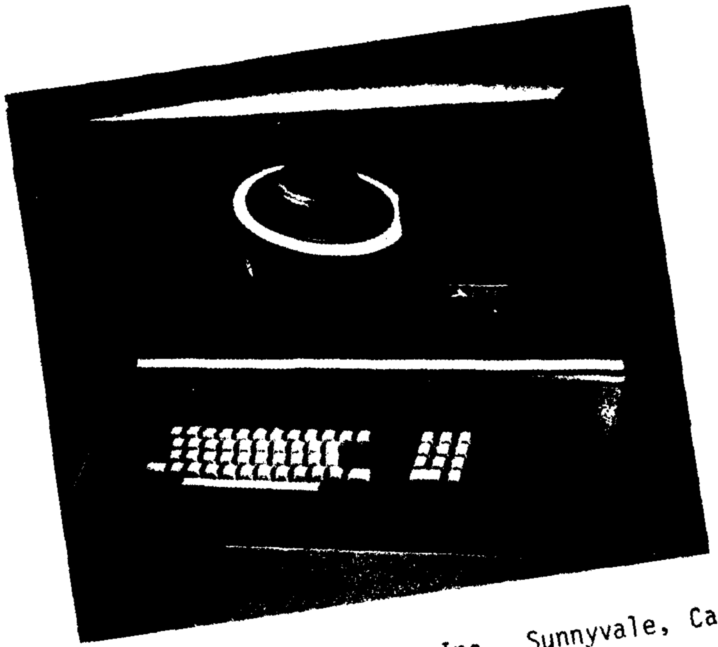


(c) Pen plotter - Soltec, Corp., Sun Valley, Calif.

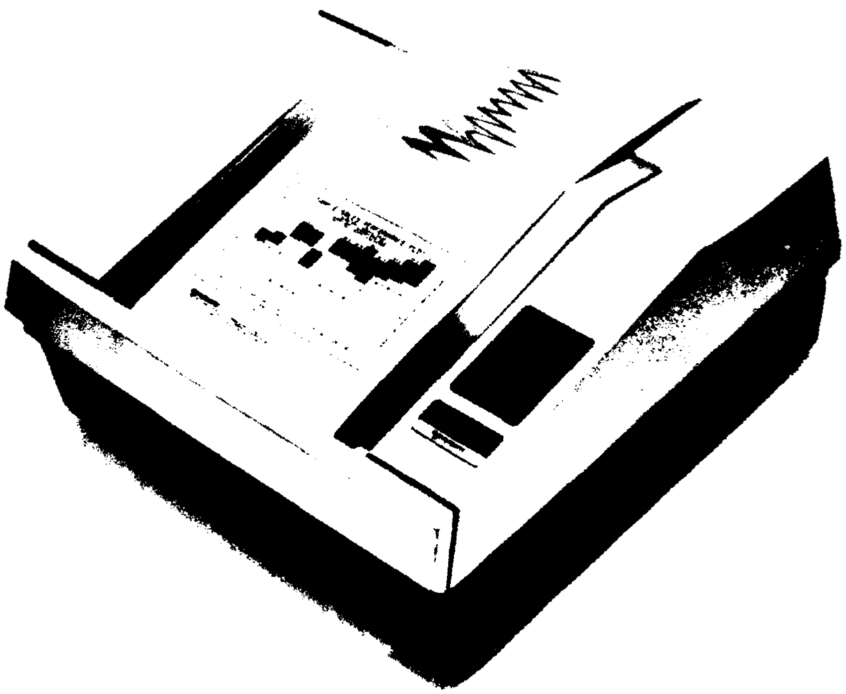


(d) Color graphics terminal - Hewlett Packard, Palo Alto, Calif.

Figure 3. (Continued).

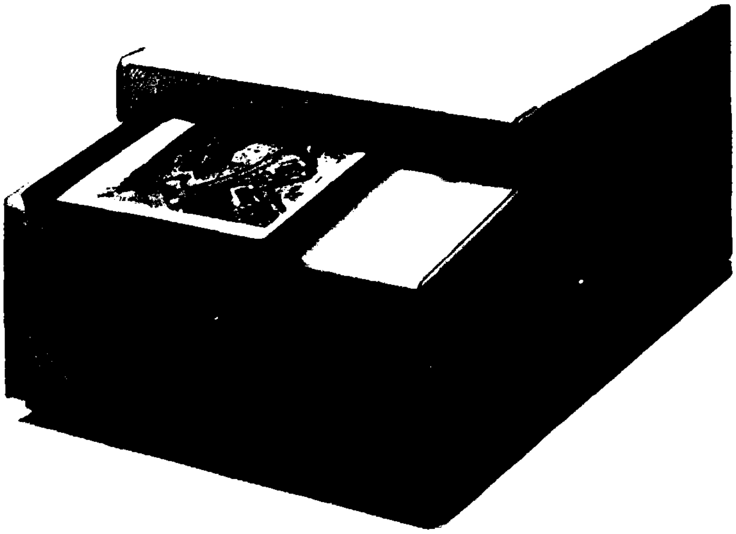


(e) Image processor, AED, Inc., Sunnyvale, Calif.



(f) Electrostatic plotter - Versatek, Santa Clara, Calif.

Figure 3. (Continued).

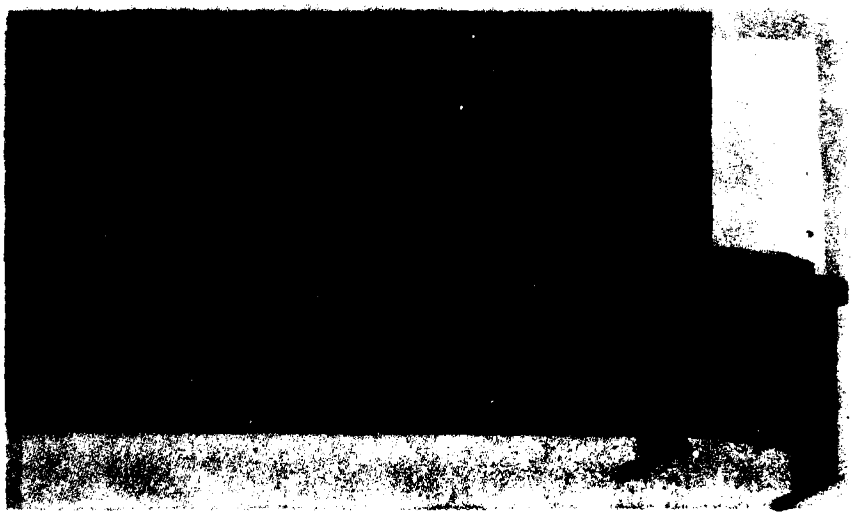


(g) Video imaging copier - Tektronix, Beaverton, Oregon.

(h) 12 grey-level plot - Tektronix, Beaverton, Oregon.

Figure 3. (Continued).

(i) Landsat Photograph - Grinell Systems, San Jose, Calif.



(j) Film recorder - Dicomed, Minneapolis, Minn.

Figure 3. (Concluded).

a. Mapping procedures:

   (1) Contouring
   (2) 3-D surfaces in perspective

b. Vector/tensor representation
c. tracer particles
d. 3-D objects in perspective
e. Hidden-line/hidden-surface algorithms
f. Shading: color, grey-level scaling, half-toning, digital printing
g. Realism: shadows, transparencies, reflections
h. Animation
i. Image processing
j. CAD (computer aided design), schematics
k. Computer generated art

A detailed discussion of these software techniques with accompanying examples follows.

III. CATEGORIZING ENGINEERING DATA

In the interest of clarity, engineering data will be categorized into two groups: real and simulated. The latter group will be further subdivided by its multidimensionality.

Real data are raw physical data: The surface of Mars, the earth's terrain, fluid flowing in a duct, airplanes, etc., where images of these subjects are obtained by some video, mechanical or electro-optical scanning device. Simulated data are artificially generated: The theoretical pressure in a wind tunnel, or the velocity flow field about a new aerodynamic concept. In this case, images are formed by executing a numerical algorithm on the computer to "model" or "simulate" some physical phenomenon, e.g., turbulence. In both cases, the data are "digitized"--images are completely represented by a series of numbers and are usually in matrix form so that each element of the digitized matrix corresponds to a picture element on a viewing screen, and the value of the element may represent the intensity or color of the image.

As an example of real data, Figure 4 illustrates the surface of Mars* where information was obtained from a video scan device and relayed back to earth for image processing. An example of artificial data is the computer modeling (Ref. 2) of the flow patterns of a gas around an aerodynamic body (Fig. 5).

Although these two types of diverse data have significantly different origins, once the data is digitized, the graphics analysis is performed on the same type of data base and graphically displayed without regard to its real or artificial nature. One of the fundamental, albeit expensive, objectives of computer graphics is to add realism to artificial simulations so that one can facilitate the analysis of data. If digitizing is thought of as "decoding" an image, then employing appropriate software image enhancement routines can be considered as an "encoding", or image generation procedure.

---

2. Feldman, Laurence A., "A Numerical Scheme for Predicting Transient Shock, Boundary Layer, and Magnetohydrodynamic Phenomenon," Arnold Engineering Development Center, Arnold Air Force Station, Tenn., Dec 1978.

*Morvec, Hans, "Surface of Mars," Artificial Intelligence Laboratory, Stanford University, Calif., Jan 1980.

Figure 4. Surface of Mars. Employing a 'distribution half-tone' algorithm on a Benson-Varian electrostatic plotter. Data from Mariner exploration. Hans Morvec, Artificial Intelligence Laboratory, Stanford University, Calif.

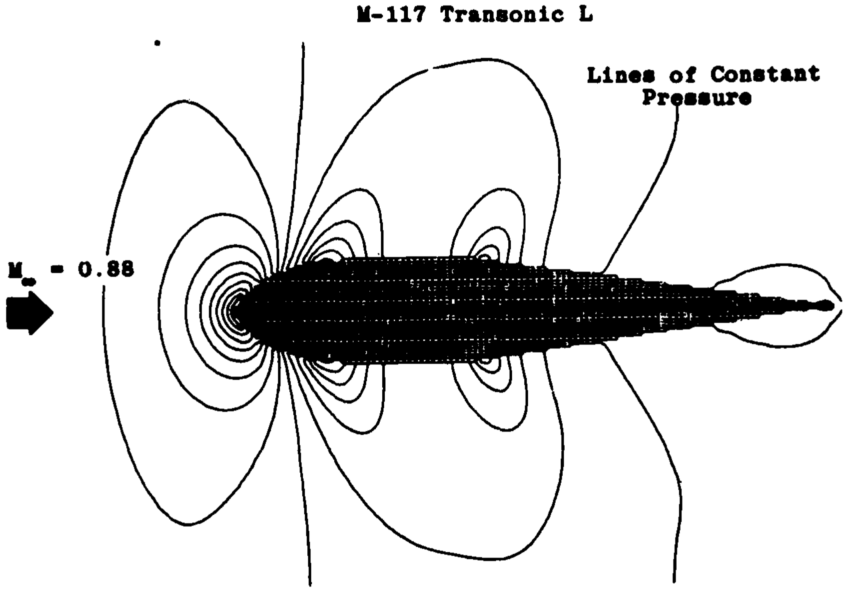


Figure 5. Aerodynamic body. Pressure contours of 2-D transonic flow about an M-117 weapon system.

In the past, real data were usually displayed by image processors and engineering simulation data by less sophisticated plotters. The present trend is toward sophisticated image processing for both sets of data.

1. DIMENSIONALITY

The analysis of numerical simulations is characterized by the number of (1) spatial, (2) independent, and (3) dependent variables. The differential equations which serve as the basis for the model dictate the number of dimensions. The aerodynamic body shown in Figure 5 has two spatial dimensions: x and y; two independent variables: x and y since $p = p(x,y)$; and one dependent variable: p or pressure. This numerical problem is two-dimensional (number of independent variables) but, if plotted as a 3-D surface map where pressure represents the z-coordinate, the calculation is still two-dimensional and the graph is three-dimensional. If the calculation was performed in three spatial coordinates and was time dependent, then the calculation is four-dimensional; yet the graphics may be characterized by 2, 3, 4 or 5 (animation) dimensions. Confusion often results over the disparity between the dimensionality of the calculation and that of the graphics.

2. NUMERICAL SIMULATION

When the flight characteristics of an aerodynamic body are investigated, a subscale prototype of the flight vehicle is designed and fabricated. This physical model is then placed in a wind tunnel which simulates the anticipated flight environment the vehicle will encounter. This testing procedure produces the necessary lift, drag and moment data to evaluate the performance of this design concept under conditions approximating true flight. Unfortunately, wind tunnel testing is expensive, (> $100,000), due not only to the cost of material, labor and operation of actual wind tunnel testing, but also to the expense of fabricating a precise prototype.

Use of the computer as a limited alternative to wind tunnel testing has received growing interest over the past decade. The computer models a variety of aerodynamic designs, environment and flow conditions--thus generating the same lift, drag and moment data, but at a significant reduction in cost.

Although the experimentalist who performs the physical experiment and the computer theorist who performs the numerical experiment may not be in full agreement as to the effectiveness of each approach, both recognize the contribution that each can make toward better understanding the performance of an aerodynamic body without having the body leave the ground. A brief discussion of just how the numerical aerodynamicist models his experiment on the computer follows to set the stage for the vital role computer graphics plays in evaluating the voluminous data this model will generate.

The physical equations governing the behavior of a fluid flowing over an aerodynamic body are commonly referred to as the equations of motion. Consider the fundamental vector equations governing the behavior of an electrically conducting compressible viscous fluid:

CONSERVATION OF MASS:

$$\frac{d}{dt}(\rho) + \rho(\nabla \cdot \vec{v}) = 0 \tag{1}$$

CONSERVATION OF MOMENTUM:

$$\rho\frac{d}{dt}(\vec{v}) + \nabla p = -\rho\vec{g} - \nabla \cdot \overset{=}{\tau} + \vec{j} \times \vec{B} \tag{2}$$

CONSERVATION OF ENERGY:

$$\rho\frac{d}{dt}(e) + \nabla \cdot (p\vec{v}) = -\rho\vec{v} \cdot g - \nabla \cdot (\overset{=}{\tau} \cdot \vec{v}) - \nabla \cdot \bar{q} \tag{3}$$

MAXWELL'S RELATIONS

$$\nabla x\vec{E} = -\frac{\delta}{\delta t}(\vec{B}) \tag{4}$$

$$\nabla xB = \mu_p\vec{j} \tag{5}$$

EQUATION OF STATE:

$$p = p(\rho, i) \tag{6}$$

where the dependent variables ρ, v, p, g, c, j, B, e, q, E, i, are, respectively, density, velocity vector, pressure, gravitational vector, viscous stress tensor, electric current density vector, magnetic induction vector, energy/mass, heat flux vector, electric field vector, and internal energy/mass. Supplementing these equations with constitutive relations of stress-strain, current-electric field, and heat flux-temperature, accompanied with appropriate boundary conditions, e.g.,

$$\vec{v}\cdot\vec{n} = 0 \text{ (wall)} \quad \vec{E}x\vec{n} = 0 \text{ (electrode)} \quad \vec{j}\cdot\vec{n} = 0 \text{ (insulator)} \tag{7}$$

the state of the fluid is completely described in space and time. The solution of these equations yields $\rho(x,y,z,t)$, $v(x,y,z,t)$, $e(x,y,z,t)$, $p(x,y,z,t)$ and $j(x,y,z,t)$ or any other combination of thermodynamic, gas-dynamic, or electrical quantities derived from these parameters, such as dynamic pressure, drag coefficient, or induced electric field.

To solve the nonlinear partial differential equations, a discretization procedure is used to divide the coordinate system (e.g., Cartesian) into cells of dimensions; and each cell has associated with it a volume, mass, density, velocity components, energy, pressure, etc. The boundary of the entire grid is also defined by cells which reflect the mathematical boundary conditions of the dependent quantities under study.

The discretization methods commonly employed are known as finite element, finite difference, or Fast-Fourier transforms, where the differential terms in the system of Equations 1-7 are discretized into finite terms, e.g.,:

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+\frac{1}{2}} - u_{i+\frac{1}{2}}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \tag{8}$$

and then reduced to a set of either algebraic expressions or implicity linked expressions of the dependent variables. Employing a particular marching procedure in both space and time, the values of u,v,w,e and p are then evaluated at i,j,k nodal points of the grid and at time t, where n is the cycle or marching index for time. If the grid is frozen in space, then the calculation is Eulerian; if the grid is tied to mass and displaced or deformed in time, then the grid is Lagrangian (Fig. 6).

An example of a Lagrangian calculation is shown in Figure 7, where a conventional armament deforms as the detonation wave propagates through the explosive (Ref. 3). Figure 5, flow around an aerodynamic body, is an example of an Eulerian calculation.

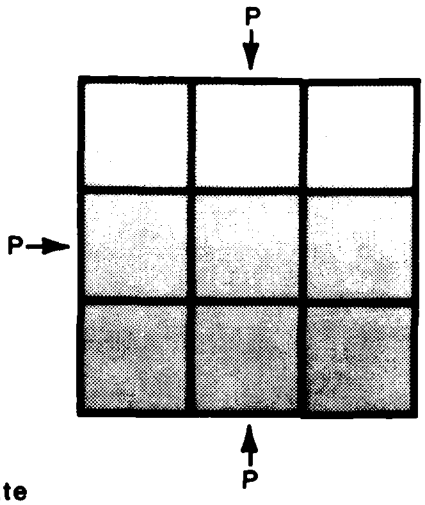
Typical grids consist of from 1000 to 100,000 cells in two- or three-dimensional grids and, for transient calculations, from 100 to 500 time iterations. Since each cell contains at least five primitive quantities (ρ, v, e), the amount of data for a typical simulation ranges from 10**3 x 10**2 x 5 (one-half million numbers) to 10**5 x 5 x 10**2 x 5 (250 million numbers). In addition, the cost of simulation runs (approximately 2 ms/cell/cycle--CDC 6600 equiv) varies from 1000 x 100 x 0.002 (200 s) to 100,000 x 500 x 0.002 (30 h).

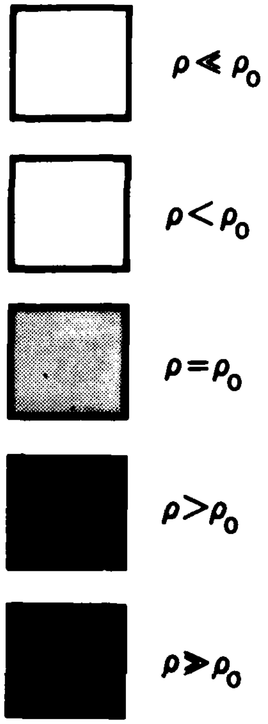
The task at hand then becomes one of not only analyzing a data file of a half million to 250 million numbers (generated by a run costing roughly $10 to $10,000 of equivalent CDC 6600 time), but also performing the analysis rapidly, still producing comprehensive data, easy to understand and communicate--a role well suited for computer graphics. In the ensuing sections, numerous examples graphically display results of models defined by Equations 1-7.

Figures 1, 5, 7, 11, 12, 16, 18, 19, 24, 25, 27 and 32 represent solutions to models of these equations and include such diverse applications as: magnetohydrodynamic (MHD) energy conversion, lasers, combustion, compressible flow in a wind tunnel, external flow about an aerodynamic body, and turbulence.

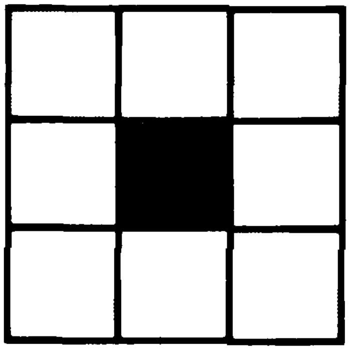
---

3. Feldman Laurence A., "Computer Animation," AFTAL-TR-73-174, Air Force Armament Laboratory, Eglin AFB, Florida, Aug 1973.





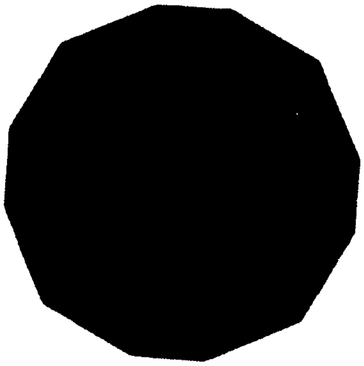
(a) Initial State



(b) Eulerian Grid (fixed volume)

$\bar{r}_{i,j} = \text{constant}$

$m_{i,j} = m_{i,j} + \Delta m_{i,j}$



(c) Lagrangian Grid (fixed mass)

$\bar{r}_{i,j} = \bar{r}_{i,j} + \bar{v}_{i,j} \Delta t$

$m_{i,j} = \text{constant}$

Figure 6. Eulerian/Lagrangian Coordinate System.

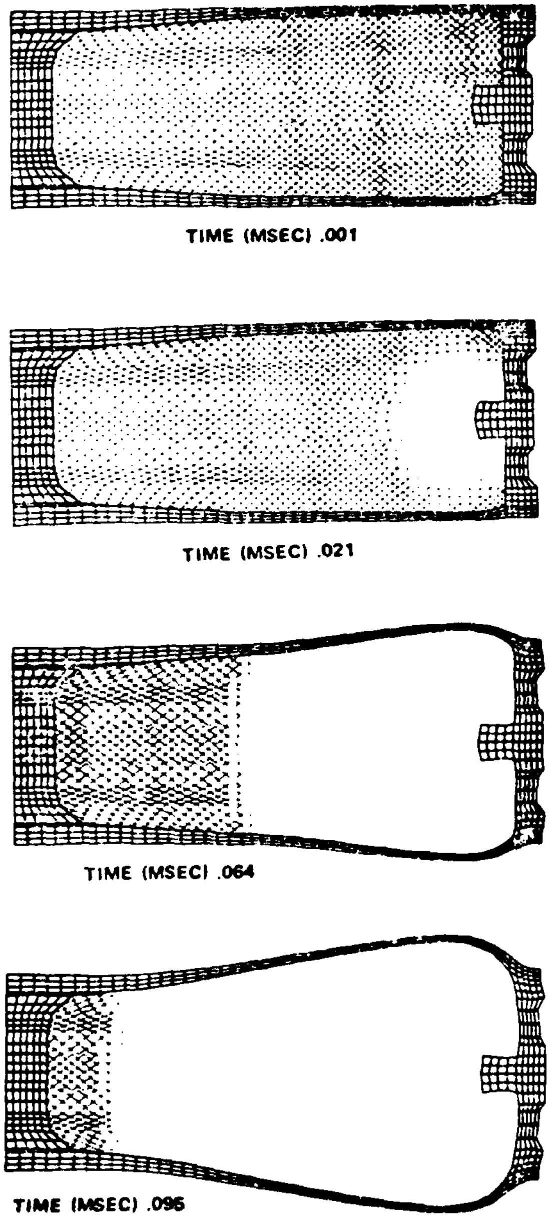


Figure 7. Lagrangian deformation. A detonation wave propagates outward causing deformation of the casing.

IV. SOFTWARE OBJECTIVES

In writing a graphics software package, certain fundamental objectives are considered. Among these were generality, simplicity, sophistication, input consistency, modularity, multidimensionality, and computer/device independence (portability).

SCAN, a SCientific ANimation computer graphics program, was written in FORTRAN IV and reflects the philosophy embodied in these objectives. The first goal was to define, develop, and incorporate all the building block graphics routines necessary for a complete package. Since analysis of multidimensional scientific data was the overriding goals of this effort, the following seemed appropriate:

a. Develop all graphs on an image plane where all data (even two-dimensional) are characterized by a three-coordinate system and the three-dimensional data are then projected on the two-dimensional image plane.

b. Since the program was to be independent of the computer it will run on (IBM, CDC, Hewlett Packard, DEC, etc.), standard FORTRAN was employed (since it is the standard engineering computer language) to give the program portability.

c. In making the program graph-device independent, standard plot calls were made in the FORTRAN program and emulators were included to convert plot calls to any particular plot device employed (Fig. 8).

d. Since the same data input may be graphed in a variety of ways, all plot packages had to accept a uniform data input file.

e. Graphs are produced on the most primitive black/white plotter or printer to the most sophisticated color image processor.

f. The makeup of the program is a number of smaller autonomous programs, each performing an independent task. Thus, to reduce memory requirements, the program is easily segmented, overlaid, or specially tailored to reduce or eliminate unnecessary software, especially for operation on a microcomputer.

g. An obvious desirable feature of any program is simplicity with the sophistication required to meet challenging applications. So in this program, attention is given to end user operation. A notated complex algorithms are included to capture the significance of the data.

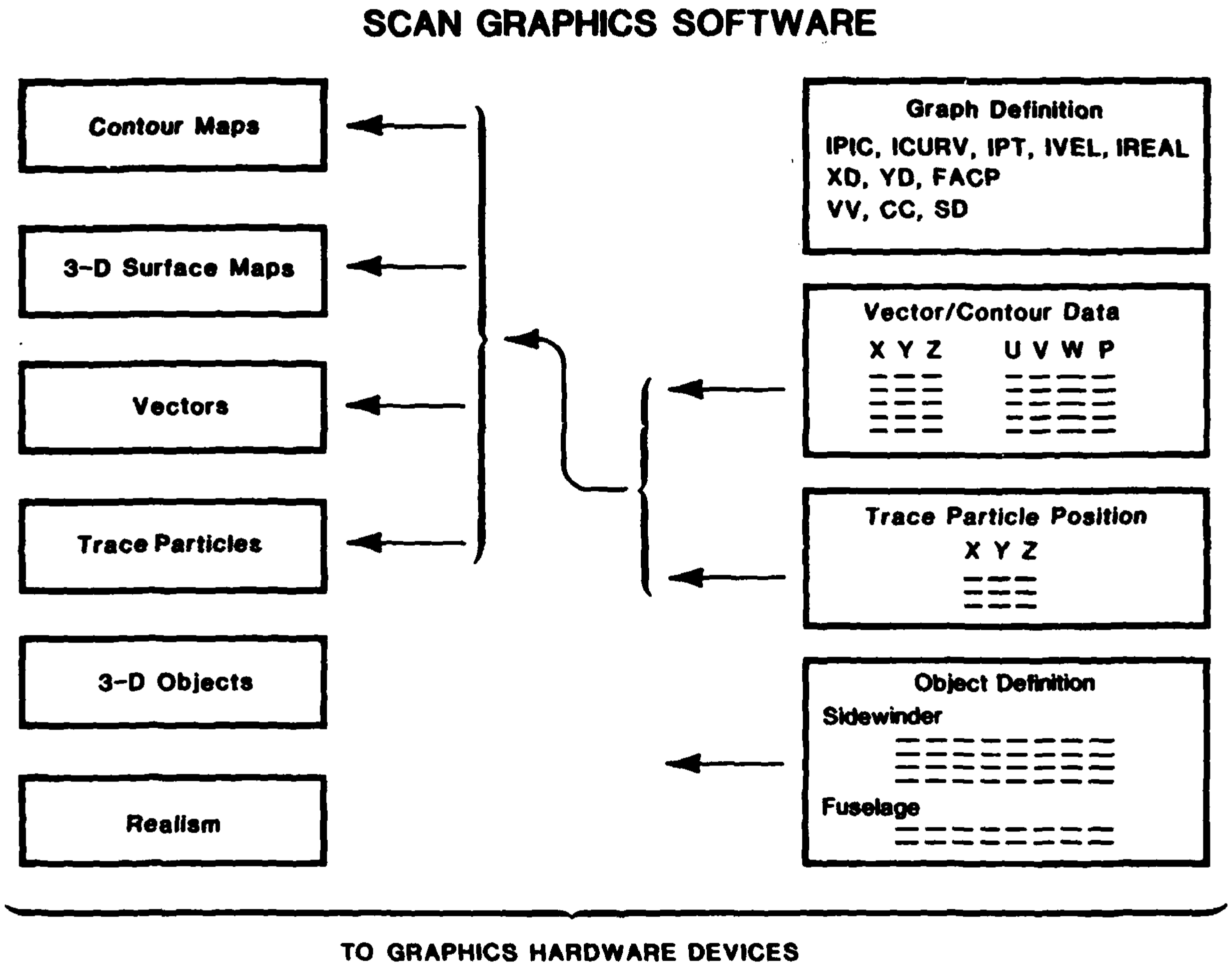


Figure 8. Device independence. "SCAN," a complete scientific graphics package is written in standard FORTRAN with plot emulators creating "portability" and "device independence."

Demonstrating software techniques to treat diverse data, Table 2 contains a list of 25 plots with their source, type of data, plot technique, graphics device, computer and the CPU time required to generate the plot.

TABLE 2. VECTOR AND RASTER SCAN GRAPHS

No.	Title	Source	Data	Plot Technique	Vector Raster	Graphics Device	Computer	Time
2	Advertisements	11	-	-	-			
4	Surface of Mars	2	3	8	R	2	3	30 m
5	Aerodynamic Body	1	1	1	V	1	2	10 s
7	Detonation Wave	12	1	-	V	10	8	5 s
10	Shock in a Nozzle	1	1	1	V	1	1	10 s
11	Turbulent Flow in 3-D	5	2	1	V	3	1	1 m
12	Energy Waves	6	1	2	V	3	1	10 s
13	Space Shuttle in Perspective			4	V			
14	F-16 Aircraft	9	4	4	R	5	1	45 s
16	Electric Current Density	10	1	1/5	V	1	2	10 s
18	Vorticities in Turbulent.Flow	5	2	5	V	3	1	2 m
19	Trace Particles in a Combustion Chamber	6	1	6	V	3	1	5 s
20	SCOTT at 2	-	4	7	R	6	1	20 s
21	Raster Scan of Boy's Face	-	4	8	R	2	1	5 m
22	Oblique View of F-16	9	4	4	R	5	1	20 s
24	Subsonic Flow in a Wind Tunnel	1	1	1/2/3	R	5	2	10 s

TABLE 2. (Continued)

No.	Title	Source	Data	Plot Technique	Vector Raster	Graphics Device	Computer	Time
25	Gasdynamic Behavior in Laser Device	4	1	1/2/3	R	5	1	1 m
26	Laser Vectors in Color	4	1	5	R	5	6	10 s
27	3-D Turbulence	5	2	6	R	5	1	30 s
28	Schematic of MHD Generator	6	-	9	R	5	1	15 s
29	Realism	3	-	10	R	11	3/4	80 m
30	Molecular Structure	8	-	10	R	12	6	?
31	Plasma Flow	13	1	1/2	V	1	2	20 s
32	Animation of F-16 Roll	9	4	4	R	4	6	10 s
33	Real Time	11	-	10	R	9	-	-

SOURCES

1	Arnold Engineering Development Center, Arnold AFS, Tennessee
2	Artificial Intelligence Laboratory, Stanford University, California
3	Bell Laboratories, Holmdel, New Jersey
4	Chemical Laser Branch, Air Force Weapons Laboratory, Kirtland AFB, New Mexico
5	Dept. of Mechanical Engineering, Stanford University, California
6	High Temperature Gasdynamic Laboratory, Dept. of ME, Stanford University, California
7	National Aeronautics and Space Administration, Ames Res. Ctr., California
8	Lawrence Livermore National Laboratory, Livermore, California
9	Revell Toy Company, Inc, Venice, California
10	University of Tennessee Space Institute, Tullahoma, Tennessee

TABLE 2. (Continued)

11	Evans Sutherland, Salt Lake City, Utah
12	US Air Force Armament Laboratory, Eglin AFB, Florida
13	Dept. of Aerospace Engineering, Auburn University, Alabama
	DATA
1	Numerical Simulation/Finite Difference
2	Numerical Simulation/Fast-Fourier Transforms
3	Video Scan
4	Hand Digitization
	PLOT TECHNIQUES
1	Contouring
2	3-D Surface Mapping
3	Color/grey-level Scaling
4	3-D Objects in Perspective
5	Vectors
6	Trace Particles
7	Digital Shading
8	Error Distribution Half-toning
9	Schematic
10	Realism
	GRAPHICS DEVICES
1	CalComp
2	Benson Varian Electrostatic Plotter
3	Versatek Electrostatic Plotter
4	Tektronix 4011
5	Tektronix 4027 Color Terminal
6	Line Printer
7	Advanced Electronic Development 512 Image Processor
8	Tektronix Hard Copy Device

TABLE 2. (Concluded)

9	Evans Sutherland
10	Friden
11	Ikonas Image Processor
12	Dicomed Film Recorder
	COMPUTER
1	HP-1000
2	IBM 370
3	PDP-11
4	VAX
5	CDC7600
6	Cyber 176
7	Cray 1
8	CDC 6600

V. BUILDING BLOCKS OF A GRAPHICS SOFTWARE PACKAGE

The two types of plot techniques: line plotting (wire frame) and raster scan (matrix of dots) will now be discussed.

1. GRAPHICS TOOLS FOR LINE PLOTTING

Engineering data have been traditionally represented through line plotting techniques: contouring, three-dimensional surface mapping (topography), schematics, vectors, trace particles, and three-dimensional objects in perspective. A discussion and examples of these techniques follow. New innovative techniques for enhancing these computer graphs will also be investigated.

a. Contouring--A contour plot is typically, although not necessarily, confined to two independent coordinates (x,y). A dependent parameter, e.g., pressure, is known as a function of x,y varying from some minimum value Pmin to some maximum value Pmax. Dividing this interval (Pmin, Pmax) into n subintervals: (Pmin, Pmin+1,.......,Pmax), lines of constant values are then plotted. To complete the contour map, each curve is annotated with the value of constant pressure it represents. The advantage to a contour plot is that it takes a large set of data and extracts only the salient features which depict trends in these data. The disadvantage is that the direction of the gradients is determined from annotation, but when contours converge, as is the case in regions where strong gradients occur, annotation becomes impossible and gradient direction information is lost. The next section deals with using color to enhance this technique and avoid the annotation problem.

An explanation of the working of the contour algorithm is as follows: consider an x,y grid of Imax by Jmax cells (Fig. 9). Treating each cell individually, when the interval in pressure from one nodal point to an adjacent nodal point contains values of pressure lying within (Pmin,.....,Pmax), a linear interpolation is performed and straight lines are drawn between end points of constant contours (e.g., 1.2, 1.3 and 1.4). When all cells are treated, contour segments automatically concatenate and contours close except at the boundary. Contour plots require little CPU time and are rapidly plotted on all plot devices. Figure 10 depicts pressure contours in a nozzle. The coalescing contours indicate shock formation (Ref. 2). Contours can analyze three-dimensional data as shown in Figure 11, where orthogonal slices of surfaces parallel to the x, y and z coordinates are plotted. Contours of

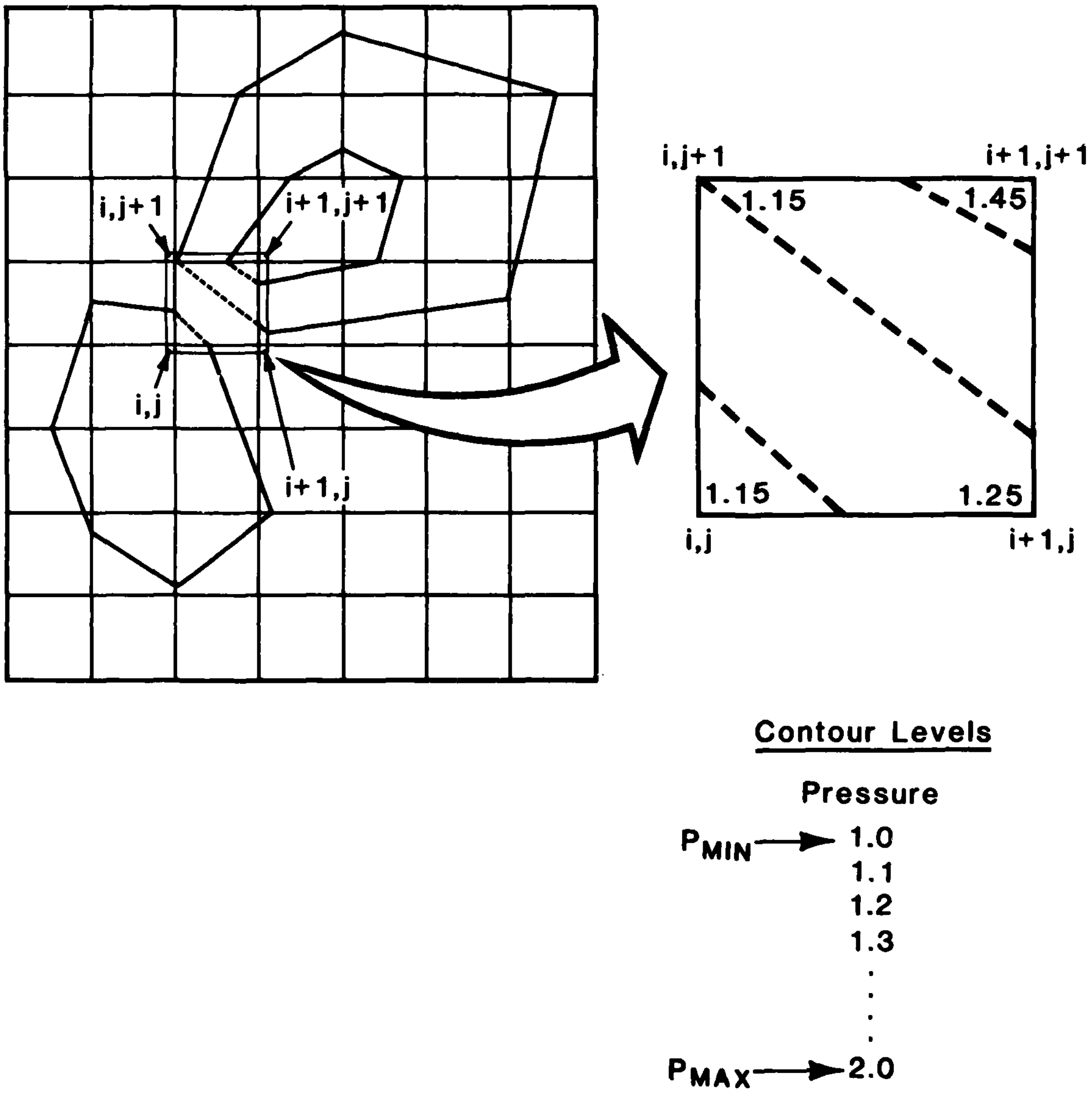


Figure 9. Contouring. A simple algorithm for 2-D plotting.

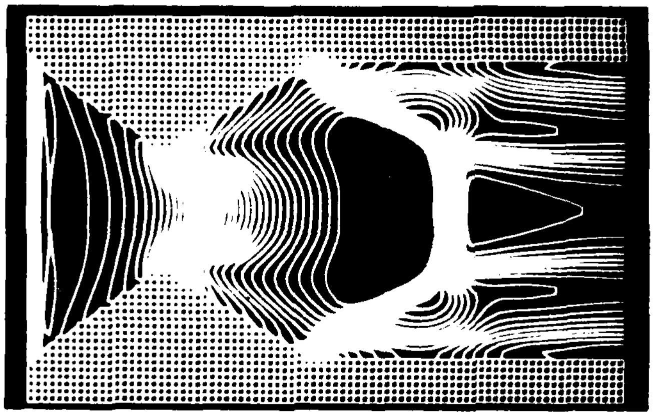


Figure 10. Shock in a nozzle. Pressure contours in a nozzle. Coalescing contours indicate a shock.

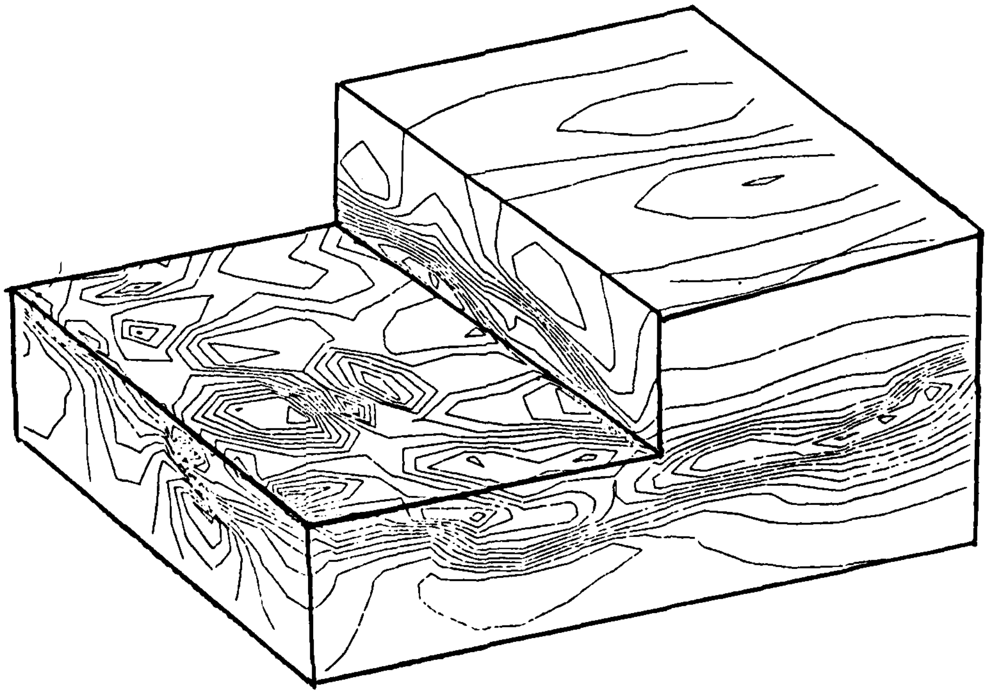


Figure 11. Turbulent flow in 3-D. Orthogonal contours of turbulent kinetic energy.

oblique surfaces can also be developed, although data must be searched and sorted in addition to applying normal contour procedures.

b. Topography--Topography or 3-D surface maps plot the same 2-D data discussed in the previous section but employ the third coordinate to represent the parameter under analysis, e.g., pressure. The advantage to this type of plot is that the direction of the gradient of pressure, as well as its value, is readily observed. The disadvantage is that a hidden line or surface algorithm is employed to "occult" lines hidden by surface closer to the viewing eye. Plots therefore require more computer time and this time increase can become substantial if the number of nodal points increases significantly. Figure 12 illustrates the transient energy distribution of waves diffusing through and reflecting against the walls of a cube (Ref. 2).

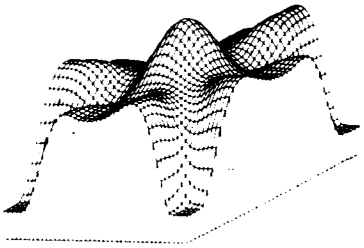


Figure 12. Energy waves. Pressure of a 2-D wave oscillating in a square; 3-D surface mapping employed.

c. Modeling a 3-D scene--Modeling a 3-D scene (Ref. 1) is accomplished in three stages: generating geometry, developing topology, and coloring surfaces. Geometry deals with measurements and dimensions of primitives (objects used in generating more complex objects). Topology deals with structure--the assembling or concatenating of primitives into objects, and objects into scenes. Surface coloring is precisely analogous to painting by numbers.

Primitive objects are composed of 3-D objects, polyhedrons, e.g., cubes, parallelepipeds, wedges, polygonal prisms, etc. The polyhedron is further characterized by the polygons* representing its faces, edges, and vertices.

---

*Another technique 'ray-casting' does not employ polygons to represent surfaces.

Scenes are viewed through a fictitious eye directed toward the object coordinate system and intersecting a plane known as the image plane. The position of the eye, image plane, and objects simulate a camera's viewing direction and aperture. Thus, if the eye is close to the image plane, the effect is a broad aperture or wide-angle lens. If the eye is located far from the image plane, the effect is a narrow aperture and telephoto lens.

A three-dimensional object, similar to 3-D surface mapping, requires an occulting procedure which treats the complete revolution of the body, as opposed to just one side of the 2-D surface as in topography. After generating a library of objects, a scenario of animation can be created where numerous objects interact on the same picture frame. CPU time is again controlled by the acculting algorithm. An example of this type of plot is the space shuttle* shown in Figure 13.

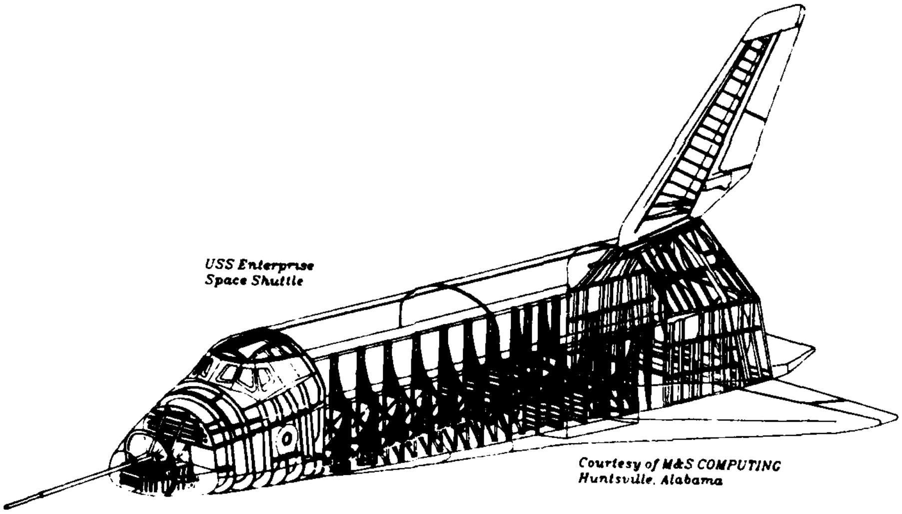


Figure 13. Space shuttle. USS Enterprise space shuttle courtesy of M & S Computing.

---

*M & S Computing, Huntsville, Alabama.

The task of converting real 3-D data into a computer input file represents one of the major obstacles in the generation of computer realism. The 3-D digitization takes on many forms, but most essentially revolve around the tedious and time-consuming chore of mechanically measuring the exterior of a 3-D body, piece by piece. An example of a prominent case in point is that of automotive design: "The initial qualification for a new automobile originates in the design center. The concepts, sketches and rendering created by styling are eventually translated into a full-size clay model. Once the design is approved, the basic specifications are determined by electrically scanning the clay model."* As an example of displaying 3-D objects, consider the F-16 aircraft shown in Figure 14.

The F-16 is composed of 123 parallelepipeds assembled from a library of 48 distinguishable parallelepipeds. Each assembled member is characterized by its x, y and z spatial location and its phi, psi and theta angular orientation (6 degrees of freedom). In addition, a scale factor and color are associated with these components. The parallelepipeds are assembled in real or object space (x,y,z) and then transformed onto an image plane (u,v).

Data for the F-16 were generated by building a scale model (Revell) and measuring with a ruler the 3-D coordinates of all components making up the entire aircraft. As an example, considering one component of this aircraft, the fuel tank is composed of three members: A, B and C where, for example, referring to Figure 15, A and B are used twice and C once. Piece B is defined by four vertices labeled (1), (2), (3) and (4) and its mirror image (1'), (2'), (3') and (4'), where the distances 1-1', 2-2', 3-3' and 4-4' are governed by dy1, dy2, dy3 and dy4. Thus, the 12 edges, 6 sides, or 8 vertices of this parallelepiped are defined by x, dy and z for points 1,2,3 and 4, where only one dy value is necessary since symmetry is assumed (about the x-z plane).

Referring to Table 3, piece (A) is initially placed into position. Adjoining this piece is (B) where the right surface of (A) coincides with the left surface of (B). The right surface of (B) then coincides with the left surface of (C), etc.

---

*Ford Motor Co., SIGGRAPH 80, Seattle, Washington.

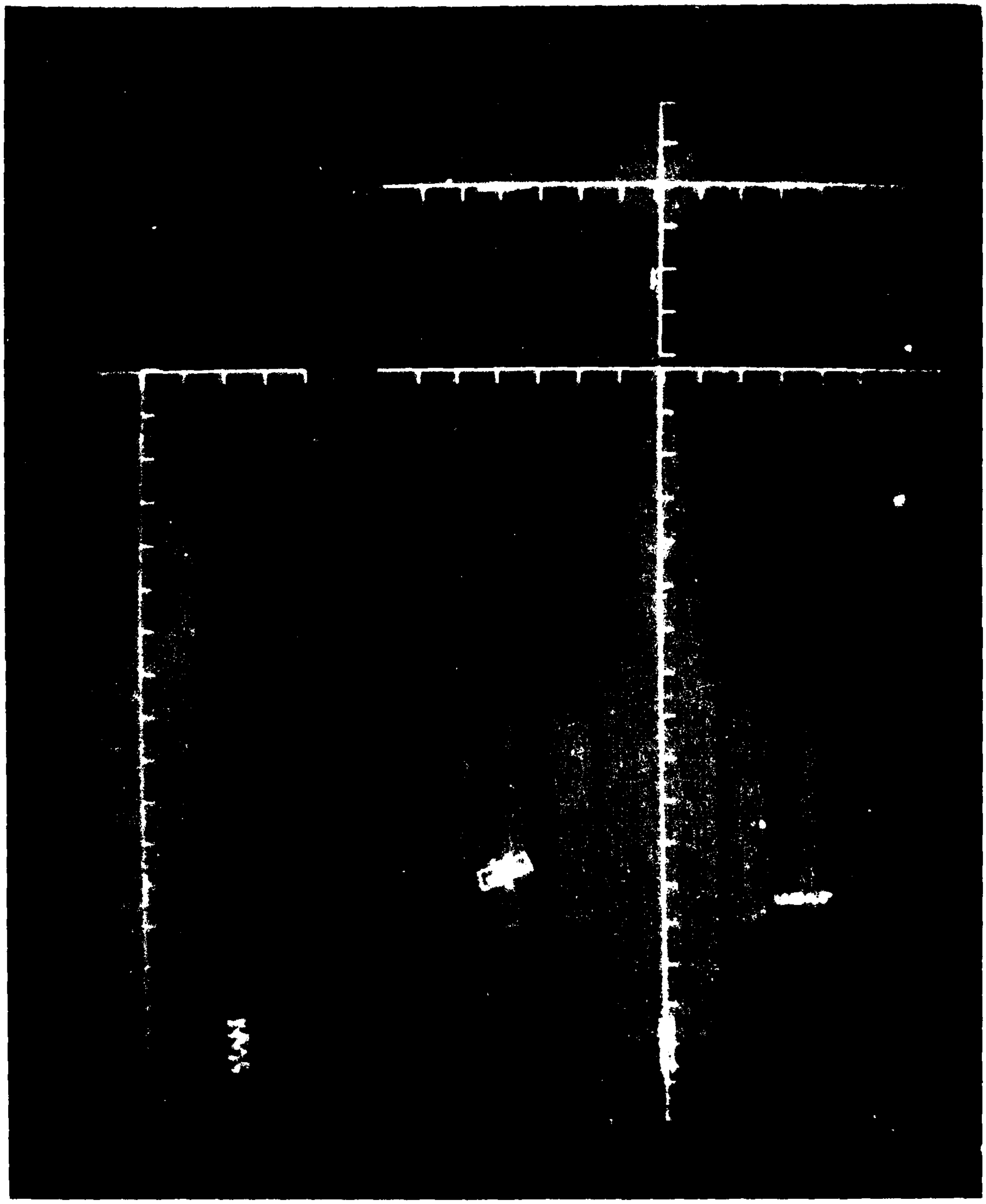


Figure 14. F-16 aircraft. Employing one data and simply 'viewing' from three different perspectives, the top, front and side views are produced.

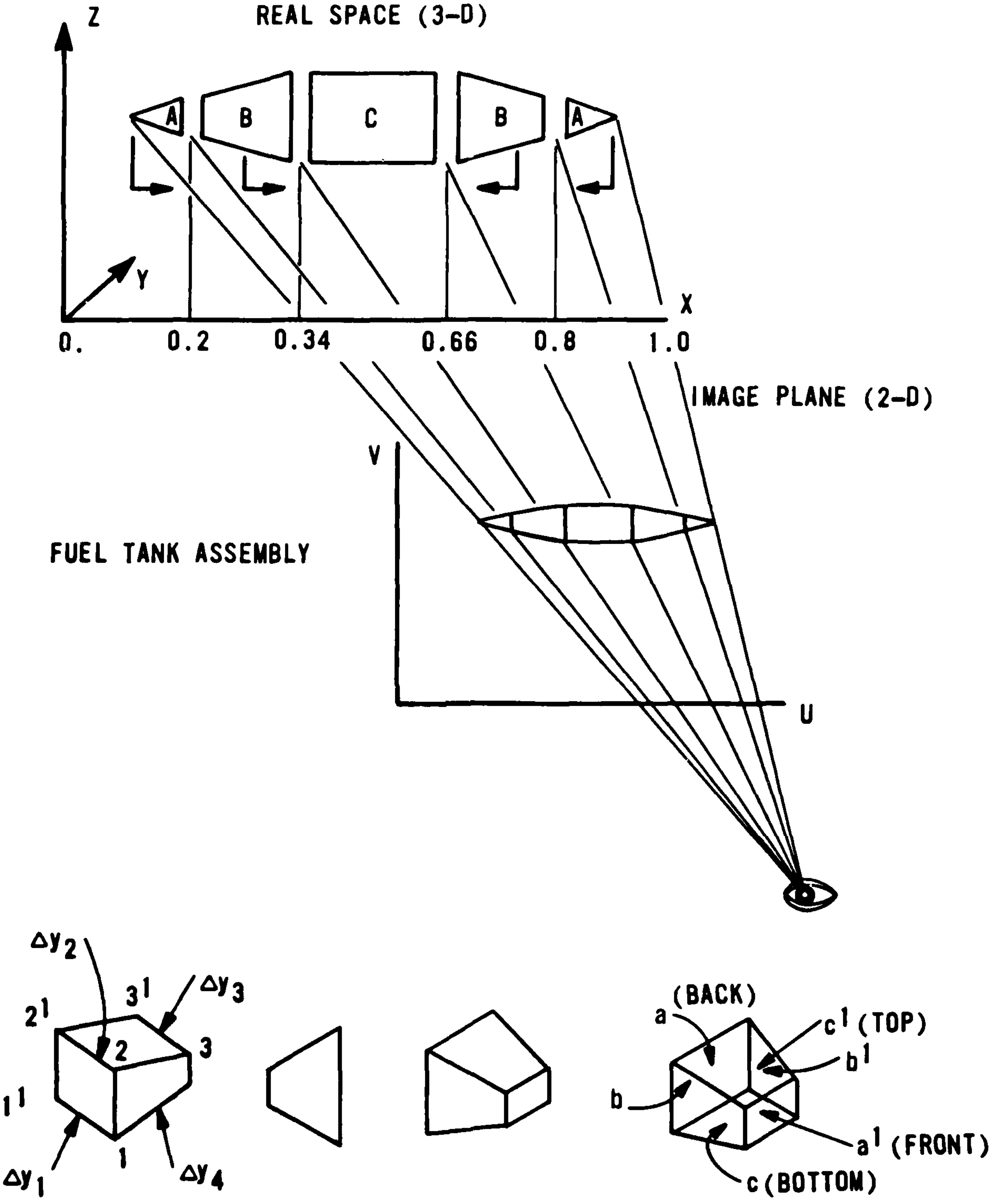


Figure 15. Fuel tank assembly. Generating a fuel tank for an F-16 by employing parallelpipeds.

TABLE 3. FUEL TANK COORDINATES

SPATIAL POSITION			ANGULAR ORIENTATION			FAC	COLOR
X	Y	Z	PHI	PSI	THETA		
0.00	0	0	0	0	0	1	10
0.20	0	0	0	0	0	1	10
0.34	0	0	0	0	0	1	10
0.66	0	0	180	0	0	1	10
0.80	0	0	180	0	0	1	10

One particular advantage in using strictly parallelepipeds is that, of the six surfaces, a maximum of three are visible at any one perspective for any parallelepiped. Assigning a, a', b, b', c, and c' to each of six surfaces respectively, by simply computing whether surface a or a', b or b', or c or c' is closer to the "eye," half of the surfaces of the parallelepiped are discarded, reducing computer time by 50 percent.

d. <u>Vectors</u>-- Vectors like vorticity, stress, heat flux, magnetic field, etc., are easily represented by arrows which indicate the direction and magnitude proportional to the length. Two-dimensional vectors are straightforward, whereas the three-dimensional counterpart requires additional treatment. First, the vectors must be displayed in perspective; secondly, a hidden line/surface algorithm may be necessary; and finally, the three components of the vector (attached to the tail) are displayed to eliminate ambiguity of vector direction. Figure 16 shows two-dimensional vectors representing the electric current density flowing between two electrodes. Three-dimensional vectors are shown in Figure 17. Note that the components attached to the tail clarify the direction of each vector. Figure 18 shows a cutaway view of a 3-D turbulent flow with vortices.*

e. <u>Tracer particles</u>--Another method for analyzing a vector field is to deposit tracer particles into the flow, allow them to assume the velocity of the position where they reside, and then observe them as they trace the path of streamlines characterizing the flow. Tracer particles deposited in the flow field of a combustion chamber are illustrated in Figure 19. Three-dimensional fields can also be represented by 3-D particles; however, one must employ (1) perspective viewing, (2) hidden line/surface algorithms, and (3) x-y, y-z, x-z plane components to avoid depth confusion. Computer time for these plots is usually not significant. One technique for resolving the depth problem when exhibiting 3-D trace particles is the kinetic depth effect discussed in the realism section.

2. GRAPHICS TOOLS FOR SURFACE PLOTTING (SHADE, COLOR)

The advent of the raster graphics device employing raster scan** has created new opportunities to develop or include realism into graphics.

---

*Cain, Alan, Dept of Mechanical Engineering, Stanford University, CA, July 1980.

**Line plotting (or vector graphics) converted to raster graphics by scan conversion.

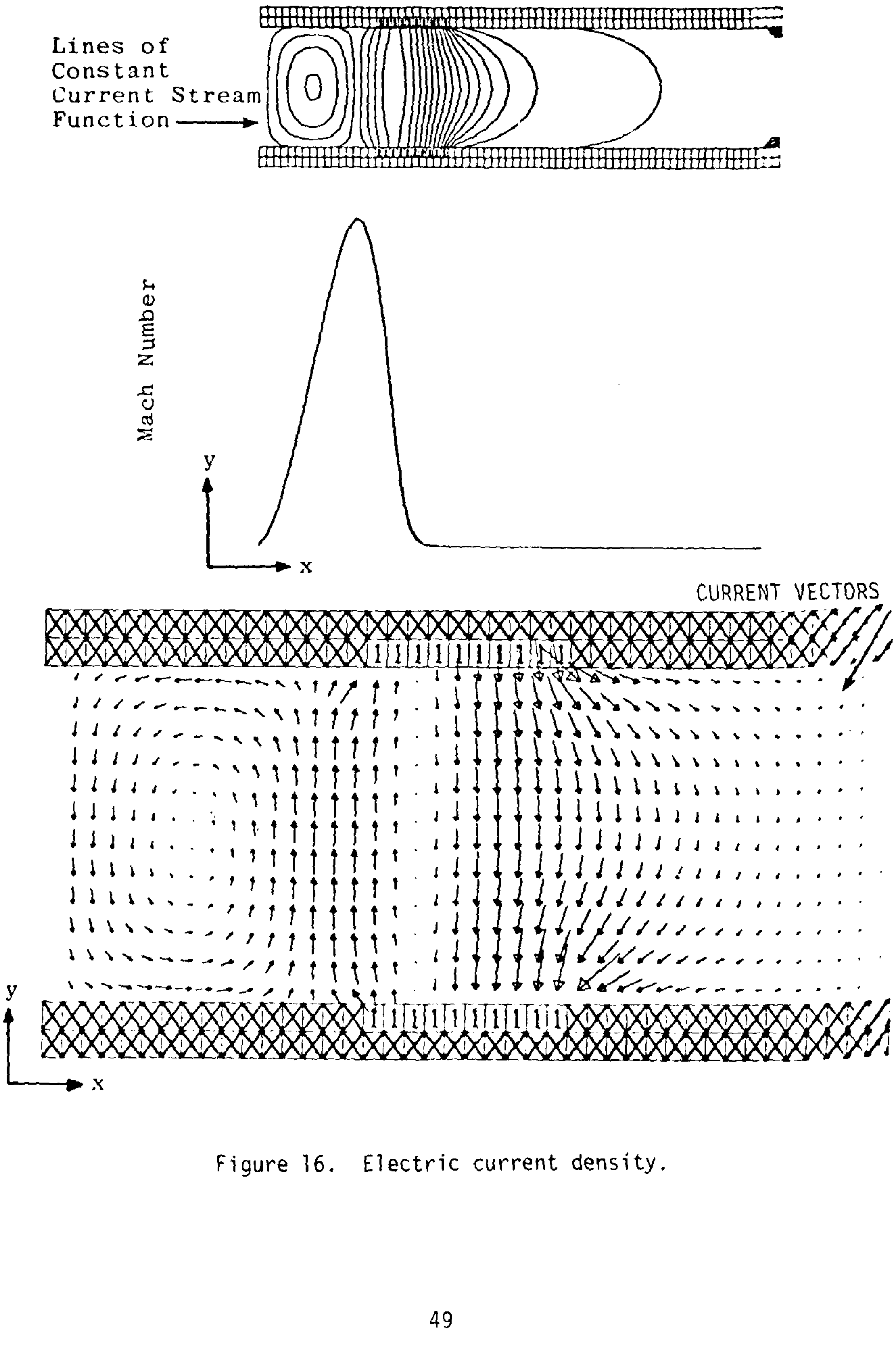


Figure 16. Electric current density.

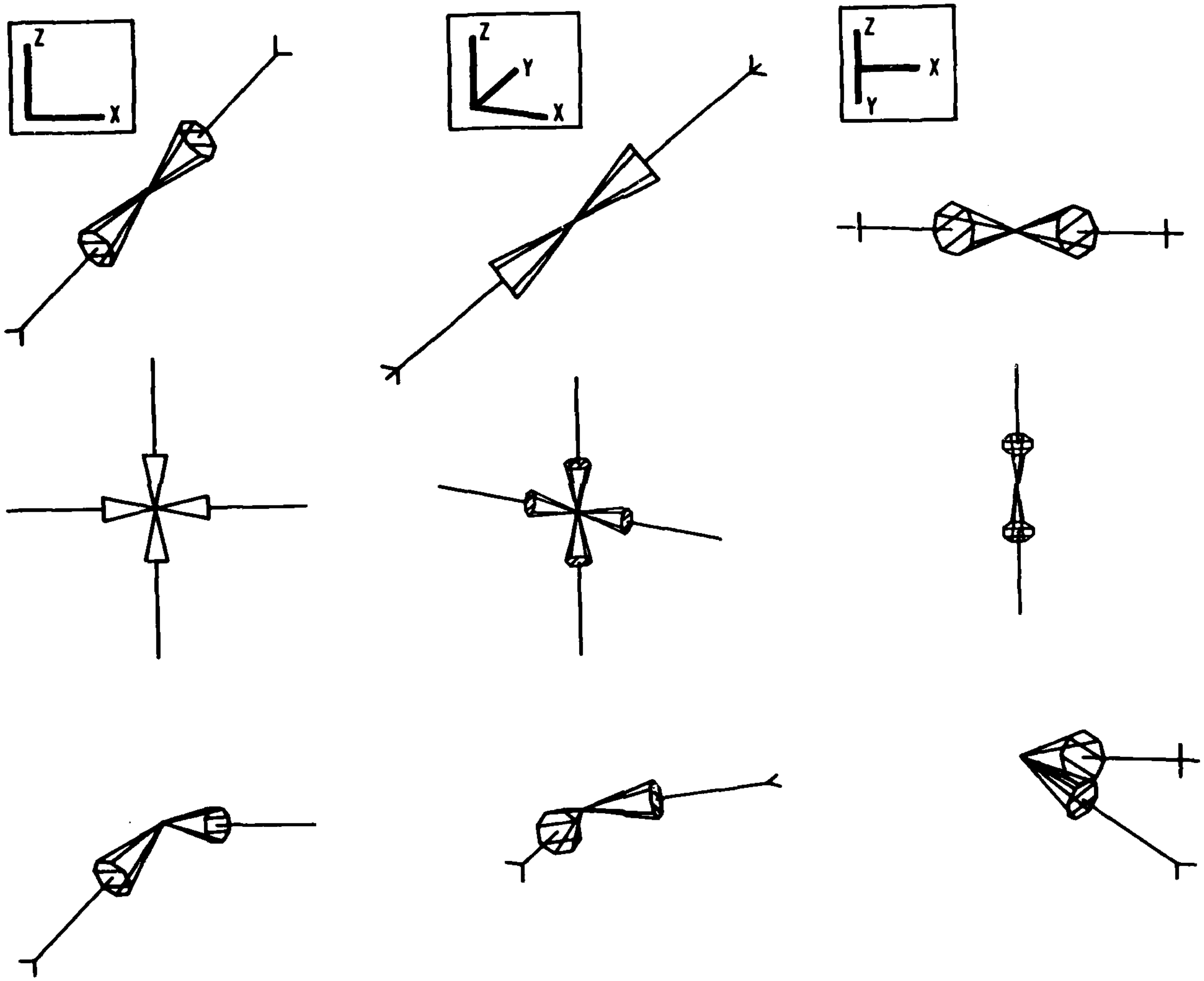


Figure 17. Vectors in 3-D. Vectors in three dimensions are plotted with components attached to the tail to clarify direction.

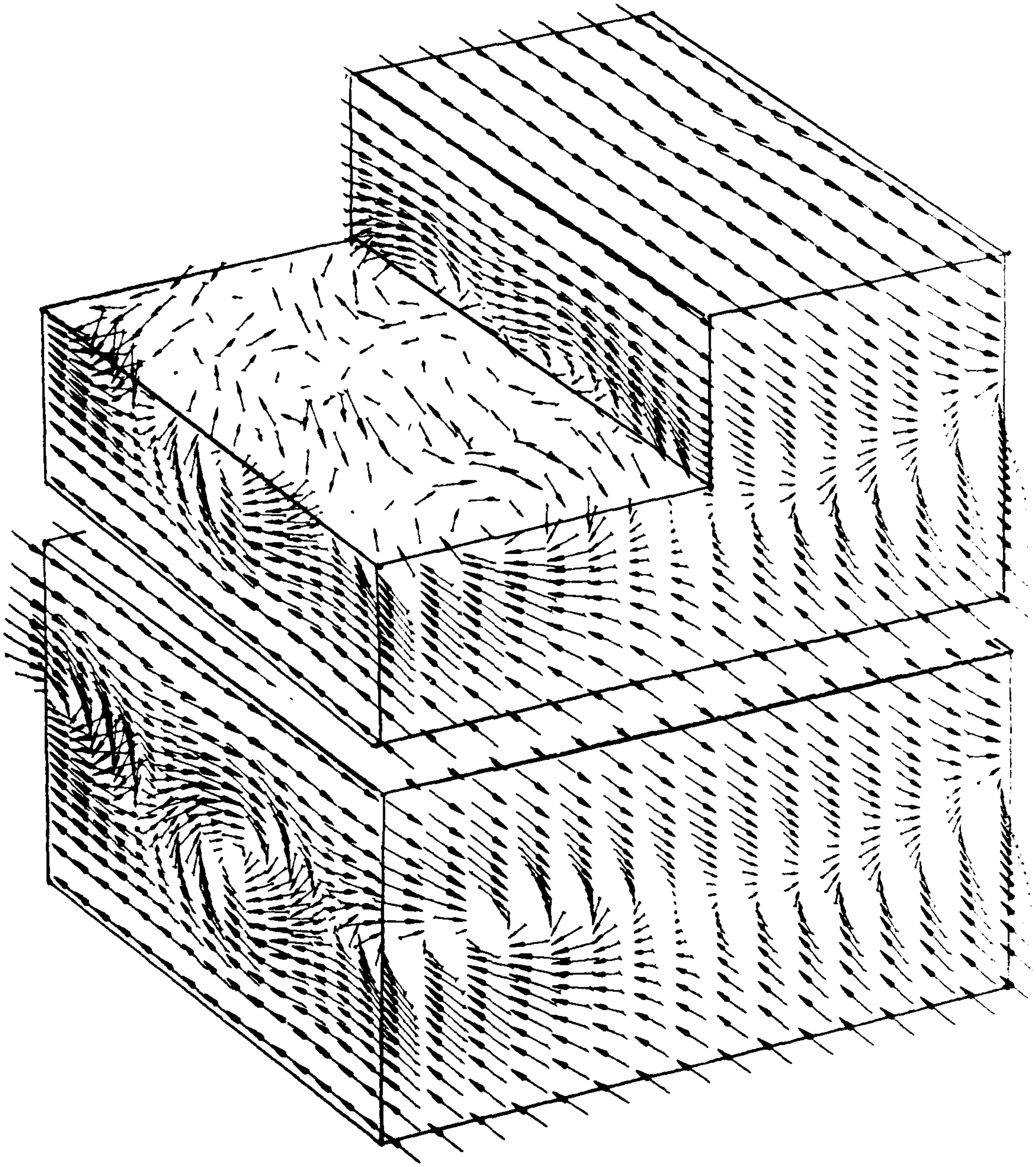


Figure 18. Vorticities in turbulent flow.

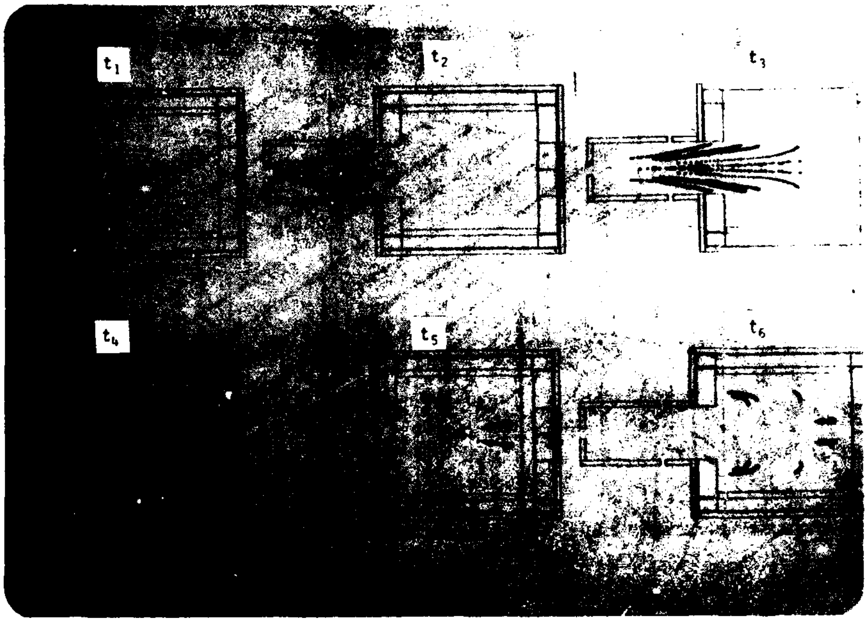


Figure 19. Tracer particles in a combustion chamber. Following the flow in a chamber by 'observing' the path of tracer particles.

Shading or coloring surfaces (usually not feasible in line plotting) not only makes data more comprehendible, but lets the visual senses perform most of the analysis and at a rate far greater than the tedious one to which the engineer has been accustomed.

Consider the algorithms discussed in the previous section: contouring, topography, 3-D objects in perspective, vectors and trace particles; next will be considered ways to enhance these plots with surface techniques. They consist of algorithms which solve the classical visible surface problem, employ shading, and generate computer realism.

a. <u>Visible surface problem</u>--The classic visible surface problem concerns the sorting of surfaces seen by the viewer and the discarding of those obstructed, occulted, or hidden from the viewer. Plotting 3-D objects in perspective involves the execution of simple 3-D transformations or matrix inversions. What makes the problem challenging is to compute the surfaces that are not visible and eliminate them. A discussion of hidden surface algorithms is beyond the scope of this report; however, a brief introduction into one of the more popular techniques is included. If a raster device is being used, the picture or frame buffer contains a matrix of dots or pixels of, for example, 512 x 512. This matrix of values is commonly referred to as the bit-map. The depth buffer or z-buffer technique requires that at each and every pixel or dot, two values be stored: the color or intensity value of the object that lies closest to the viewer and the distance of that object from the viewer. If a new object located in a region containing this pixel is closer to the viewer than the previous stored depth value, then the intensity and depth values are updated to contain the values of this new object. If the new object is found to lie further away from the viewer than the previous object, it is hidden behind that object and therefore discarded--for that pixel only. This process is repeated for each object and pixel, 250,000 of them. Since this procedure is time-consuming, a great deal of research is devoted toward improving the efficiency of hidden surface algorithms.

b. Shading--A variety of techniques for shading exist, and all are device dependent. As an example, consider a matrix of 72 x 102 elements whose values range from 1-16. A "1" indicates black, a 16 indicates white, and 2-15 indicate grey-levels. To generate an image (boy's face) on a line printer, associate one-to-one each element of the matrix to a pixel on the line printer output. In this case, a pixel represents the space allotted for each character (132 per line/60 per page) on the printer output. Grey-level is achieved as follows (see Table 4). For light grey, start with a light intensity character and increment as follows: a ".", to a "," to "/", etc. For darker greys, overprint by backspacing on the printer and using combinations of characters 0, $, X, etc., filling up the entire region of the pixel. Figure 21 illustrates the results of this mapping procedure. Contrast is adjusted by simply modifying the elements of the matrix as follows:

$$\text{Intensity} = \left(\frac{I}{16}\right)^{n} 16 \qquad (9)$$

where I is the old intensity varying from 1 to 16 and n is the contrast exponent. In Figure 20, three contrast values are shown: n = 0.8, n = 1.2, and n = 1.8.

Next, consider a raster scan device employing a one-bit frame buffer; that means each dot on the screen or the plot is either on or off, and the dot has

TABLE 4. DIGITAL SHADING

INTENSITY LEVEL	CHARACTER SEQUENCE									
1	.									
2	/									
3	o									
4	o	'								
5	+	/								
6	X	X								
7	0	'								
8	0	0	'							
9	0	0	'	'						
10	0	0	0	/	/					
11	0	0	0	;						
12	0	0	0	+						
13	0	0	0	X	-					
14	0	0	0	X	*					
15	0	0	0	X	X	X	*			
16	0	0	X	*	%	M	W	$	-	$

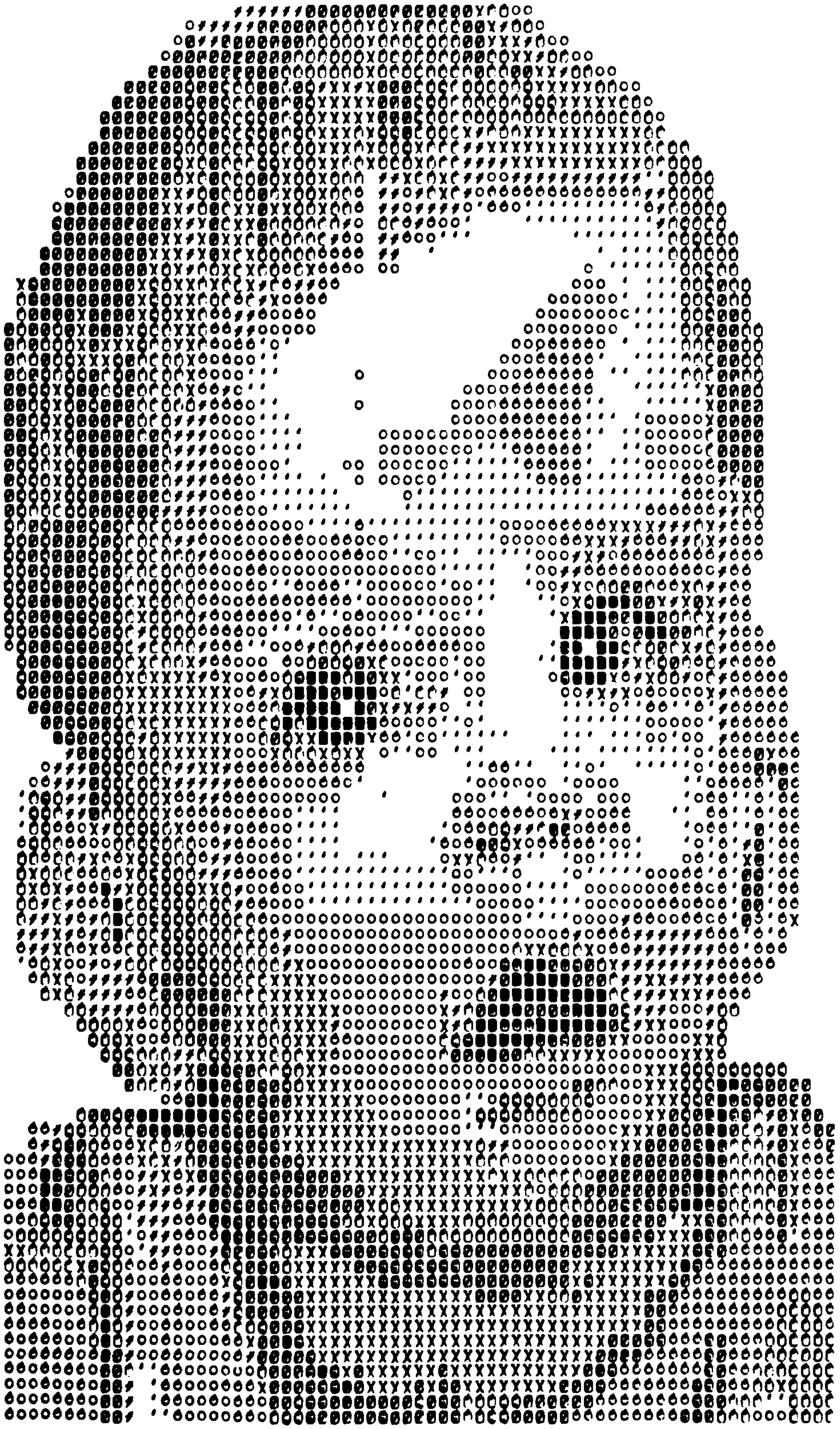


Figure 20. Scott at 2.

Figure 21. Raster scan of boy's face.

no intensity level. Employing a "half-tone error distribution" algorithm (Ref. 1) where the frequency of occurrence of dots (as in newspapers) dictates grey-level (monochromatic) intensity, the boy's face is recreated (Fig. 21). The surface of Mars shown in Figure 4 was also generated using this algorithm. Figure 21 also contains the results of using a 4-bit device with 16 intensity values. The important point is that shading is generally feasible on most devices, with some offering more in resolution and quality than others, but with all offering a tolerable picture even though the device was not necessarily designed to accomplish such a task. Line plotting devices shade by spacing lines closer together (darker) or further apart (lighter), but unfortunately do not work well because of compute and plot times involved; and the quality is inferior to the plots described above.

Employing a color graphics terminal (8 colors), the F-16 shown in Figure 22 is constructed with four colors. For the F-16 to resemble the real aircraft, a resolution of approximately 1000 x 1000 points and a minimum of 10-16 shades of grey are necessary.

Next consider the contour plots and 3-D surface maps shown in Figures 5, 10, 11 and 12. The effect of color in producing contours/topographical maps allows for rapid clarification of regions of low or high Mach number as shown in Figure 23. In Figure 24, the flow in a wind tunnel is described by color coding of the Mach number. Suppressing the colors representing supersonic flow (> Mach 1), the eye can instantly grasp regions of supersonic flow.

Vectors may exploit colors by allowing all to be of uniform size and by having color or grey-level represent magnitude (Fig. 25). Color can also be used to represent vorticity while representing magnitude by the length of the vector.

The 3-D trace particles effectively use colors by representing each point of origination of the trace particles with a different color and then following the path of the colored particles to trace streamlines. Figure 26 illustrates these colored streamlines generated from a 3-D turbulent simulation.

As a final example of color, a schematic of an MHD energy conversion device is shown in Figure 27. The plasma flow down the generator is represented by grey-level associated with the pressure inside the generator.

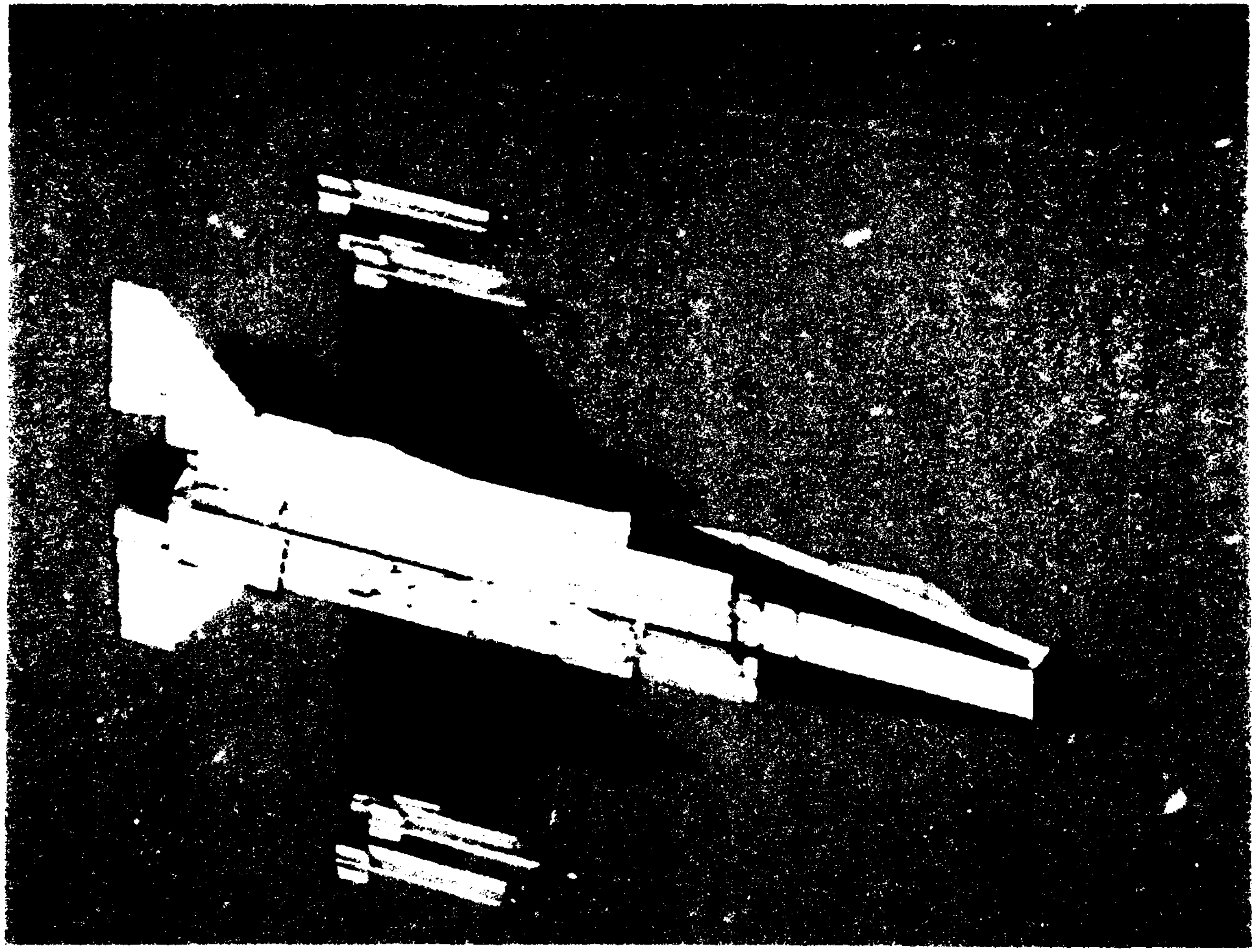


Figure 22. Oblique view of F-16.

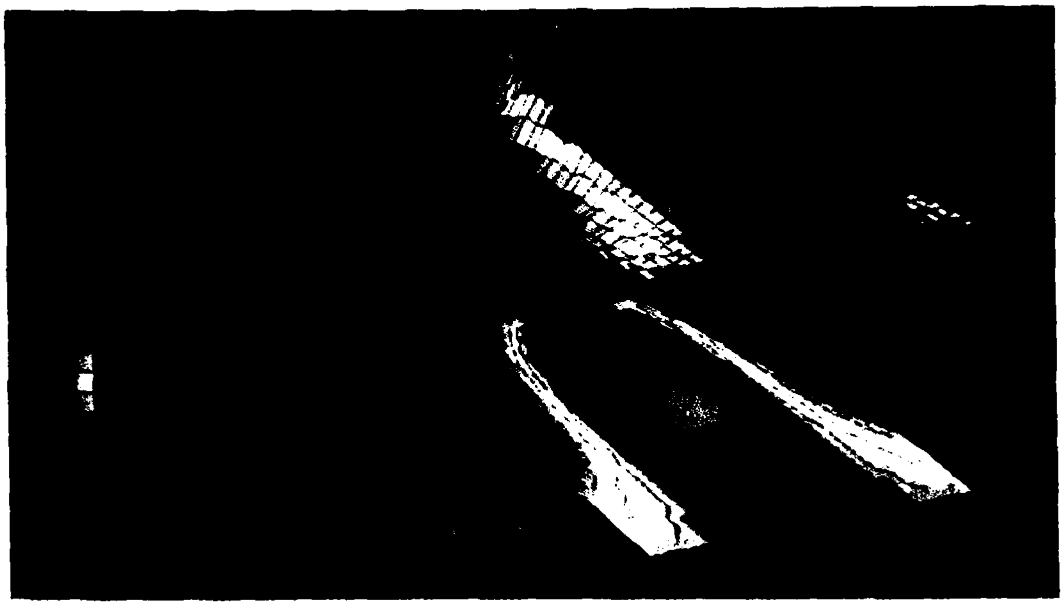


Figure 23. Laser device. Employing color to emphasize pressure regions in a contour, 3-D surface map.

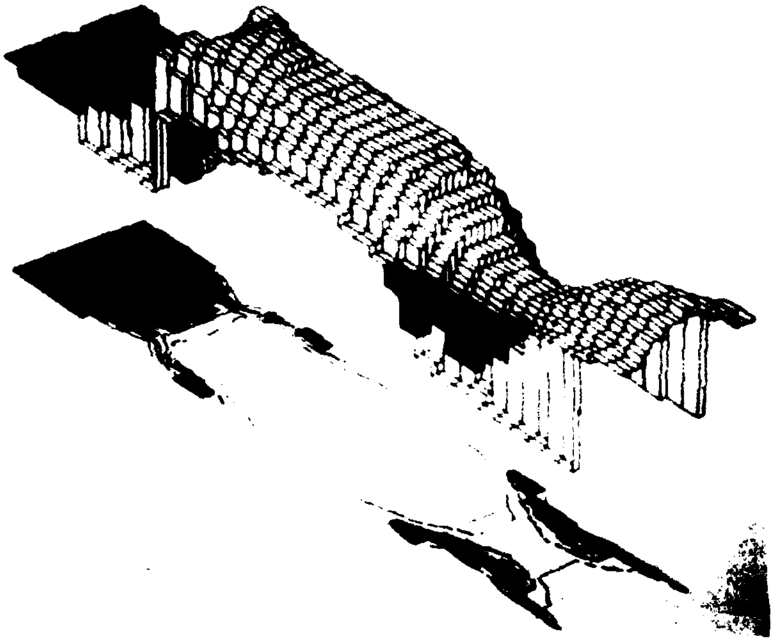


Figure 24. Subsonic flow. Employing color to highlight subsonic region in wind tunnel flow.

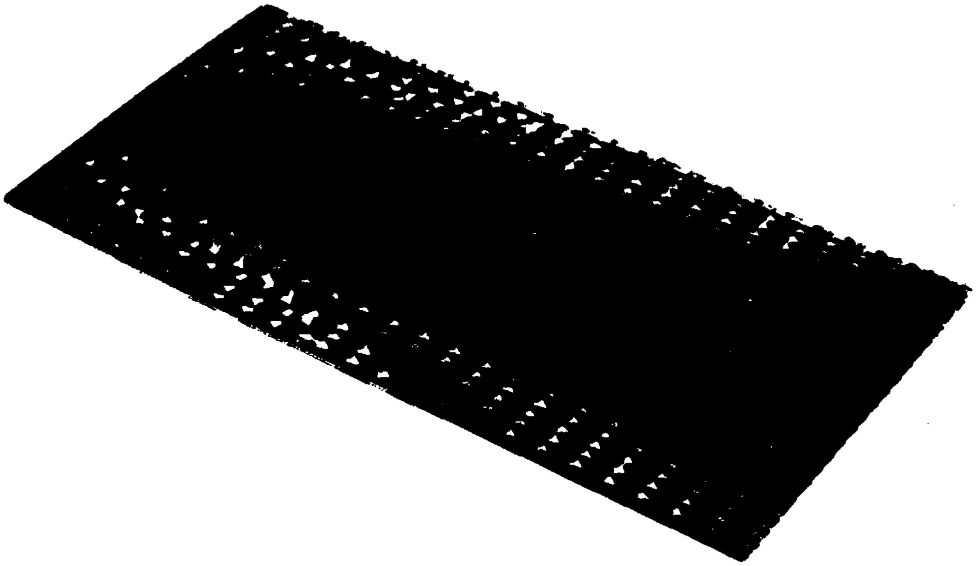
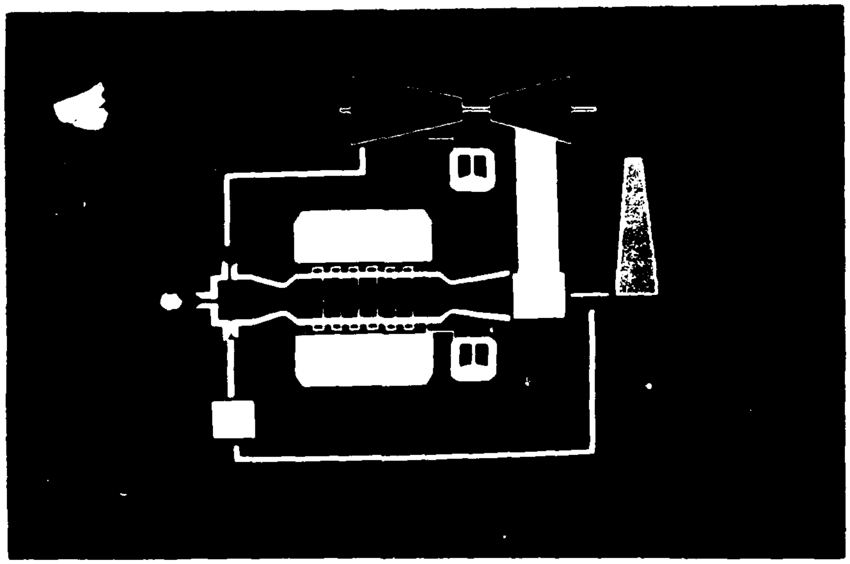


Figure 25. Vectors in color.

Figure 26. Turbulent 3-D tracer particles.



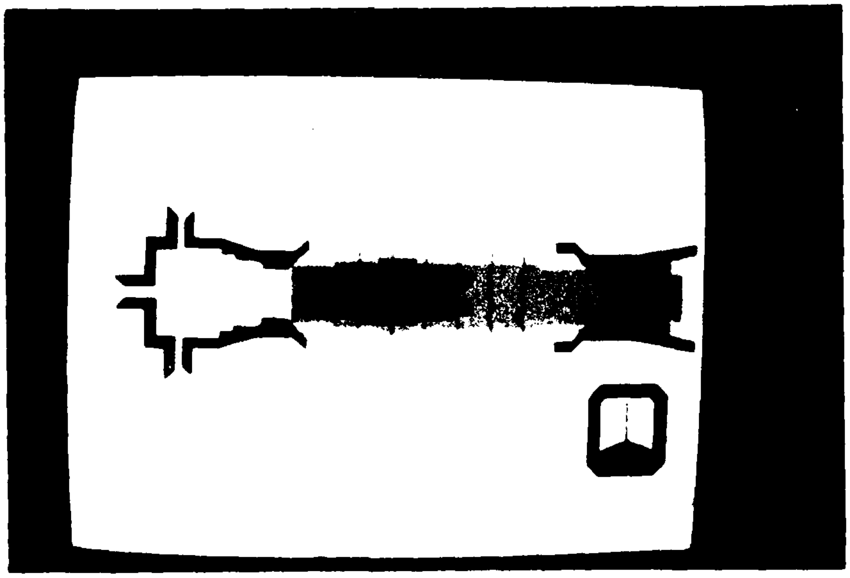


Figure 27. MHD generator.

c. Realism--Previous discussions dealt with pictures containing objects composed of a series of lines and shaded polygons, e.g., the F-16 jet aircraft, 3-D surface maps and velocity vectors. Although the graphs represent the data in a meaningful fashion, realistic visual effects such as smooth surfaces, shadows, light reflections, and transparency were absent.

Incorporating realism techniques not only eliminates the artificial caricature effect of computer graphics, but also imitates real life so precisely, in some cases, that one forgets the picture was created artificially by the computer. Success in implementing these realism algorithms depends principally on the availability of sufficient computer time and a graphics device which provides high resolution (color and pixel density) output. Algorithms that approximate light reflection, absorption and transmission, and shadows cast by adjacent objects require substantial computer time--orders of magnitude greater than that required to generate the plots presented in the previous sections. Figure 29 required 60 to 80 min of PDP-11 computer time. In addition, continuity in geometry and lighting requires enhances resolution to create smooth transitions from neighboring regions of sharply contrasting light intensity.

Figures 28 and 29 are illustrations (Ref. 4) of the degree of sophistication achieved when employing algorithms which model light reflection, transparency and shadows. Figure 29* illustrates the molecular structure. The smooth surface of the spheres can be achieved through (1) concatenation of a sufficiently large number of polygons to closely approximate the curvature of the spheres, or by (2) curved surface techniques discussed in Reference 1.

An in-depth treatment of the subject of realism can be found in Reference 1. Briefly, computer generated realism concerns three general areas: (1) the classical visible surface problem (2) representation of depth or three dimensions on a 2-D image plane, and (3) contributions of lighting, color, shadows and texture to the rendering of a picture.

The first task of plotting only those surfaces visible to the eye and occulting or discarding nonvisible surfaces in an effective manner remains one

---

4. Whitted, Turner, "An Improved Illuminations Model for Shaded Display," Communications of the ACM, Vol. 23, No. 6, June 1980.

*Max, Nelson, LLL, Oct. 1980.

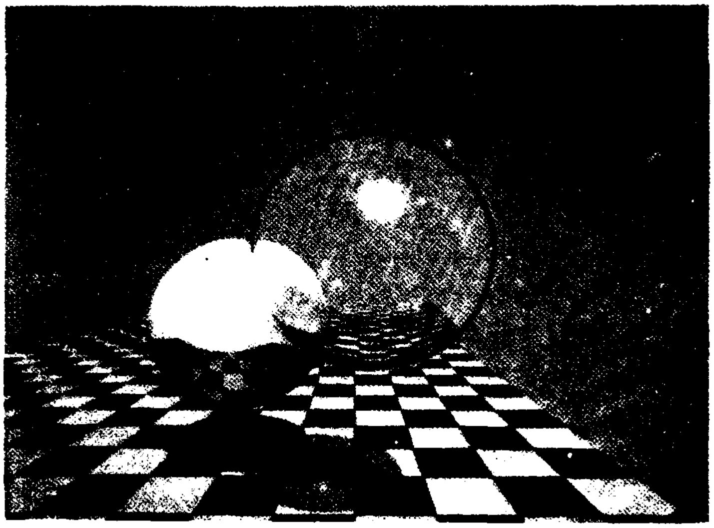


Figure 28. Realism. Curved surfaces, shadows, transparency, reflection, hidden surface. Turner Whitted, Bell Laboratories, Holmdel, New Jersey.

Figure 29. DNA molecules. Nelson Max, Lawrence Livermore Laboratory, Livermore, Calif.

of the more challenging problems in computer graphics. Depth, the second task, is generally treated by perspective projection of objects in a 3-D object space onto a 2-D image plane. Other techniques include stereoscopic viewing--employing 3-D Polaroid glasses to distinguish two separate images for each eye, and kinetic depth effect--dynamically displaying objects so that those closer to the viewer move at a faster rate than those further away.

A preliminary discussion of a shading model follows, although a more extensive introduction into the subject can be found in Reference 1. Consider a 2-D image plane containing projections of 3-D objects represented by a number of visible surfaces (polygons), where the nonvisible surfaces have already been occulted or discarded by a hidden surface algorithm. The color or grey-level of each surface is dependent on two factors: illumination (E) and radiative absorptivity (s) which governs the amount of light absorbed and the amount reflected and transmitted. The net intensity of light on the surface is equal to the net balance of light irradiated minus the light transmitted and reflected.

$$E = \sum_i s_i I_i \qquad (10)$$

where

I = the light irradiation
E = the illumination

s = the surface property determining absorptivity (0. to 1.0)

i = an index representing different irradiation sources

After I is computed for a point on the surface, a color look-up table or grey-scale is employed to determine the shading based on the input I. In the case of the F-16 aircraft, consider the sun as the source of light. The 3-D airplane configuration is already projected on a 2-D image plane where only the visible surfaces appear. Since there are approximately 120 polygons representing this airplane, each polygon must be shaded according to its natural color, the unobstructed light cast on it by the sun, and the light reflected onto its surface from neighboring surfaces of the plane. The orientation of the polygon to the sun is important because, if the surface is perpendicular to the rays of the sun, it will receive maximum light intensity; whereas, if the surface is parallel to the rays, it will not receive any light.

Expanding on Equation 9, there are three contributions to the illumination (E): (1) diffuse illumination, (2) light sources and (3) transparency effects. Thus

$$E = E_d + E_\ell + E_t \tag{11}$$

Considering diffuse illumination:

$$E_d = R_p I_d \tag{12}$$

where

I_d = diffuse illumination

R_p = reflectance coefficient (0.0 to 1.0)

Equation 11 is further divided into the primary color components of red, green and blue; but for the sake of brevity, only monochromatic grey-level will be considered.

The second term, E_ℓ *is based on Lambert's Law: "The energy falling on a* surface varies as the cosine of the angle of incidence of light."

$$E_\ell = R_p \cos i \; I_\ell \tag{13}$$

where

I_ℓ = light source illumination

i = angle of incidence

Modeling Equation 13 for specular radiation, Equation 13 becomes

$$E_\ell = \left[R_p \cos i + W(i) \cos(s)^n \right] I_\ell \tag{14}$$

where

$W(i)$ = reflection coefficient

s = angle between reflected ray and the eye of the viewer

n = shine factor (0. to 10.)

The third term of Equation 9 is the transparency effect

$$E_t = T_p \, I_t \tag{15}$$

where

Tp = transmission coefficient (0. to 1.0)

It = rear illumination

Summing Equations 11, 12 and 14, Equation 9 becomes

$$E = R_p I_d + \left[R_p \cos i + W(i) \cos(s)^n \right] I_\ell + T_p I_t \tag{16}$$

Input to the above model consists then of (1) the 3-D coordinates of each polygon, (2) a vector representing the surface normal, (3) the color of the surface, (4) radiative surface properties, (5) the light source position, and (6) the position of the observer.

VI. ANIMATION AND REAL TIME

In the analysis of engineering information, especially time dependent simulation, graphic results are shown in three modes: (1) snapshots, (2) animation, and (3) real time. A snapshot is a plot at one particular time instant and is simply one frame or one picture.

Animation is the process of recording a sequence of plots (data on each plot vary slightly from the previous one) which when shown consecutively, yield animated motion (Fig. 30). This technique is particularly effective in illustrating a transient phenomenon by shooting a picture over the physical time frame (μs, ms) until the phenomenon has acquiesced to a state of equilibrium. By employing animation, the physics under investigation is realistically analyzed. A limiting factor encountered in generating animation concerns the amount of computer and plot time required. Consider a 2-min movie which at 24 frames/s adds up to 2640 frames of film. The amount of compute and plotting time naturally depends on the amount of information on each frame; but, considering even the most modest amount of data per frame, a movie can be costly. Animation to analyze data may require only 10 to 15 s of footage; but for demonstration of results to others not intimate with the data, 10- to 15-s movies are not usually effective. The most convenient way to generate a movie is with a film recorder; but, since they are quite expensive (hundreds of thousands of dollars), the next best method (and far less expensive is to shoot directly off a CRT screen. Flicker does not pose a problem if the camera is set for long exposure.

Real time is similar to animation with the one important exception that picture generation must take place at a rate of at least 24 frames per real second. Consider an F-16 doing a roll maneuver (Fig. 31). Plotting snapshots of the aircraft at different time instants requires simply that the location and orientation of the aircraft at each time instant be computed or supplied from a data source and then sent to the graphics device at a leisurely pace. Animation of the F-16 roll requires that the location and orientation of the F-16 at specific time instants be known in order to satisfy the requirement of 24 frames/s of uniform trajectory motion. Data are then leisurely recorded on film for playback at projector speed. Real time however requires that, unlike animation, the computer and graphics system respond at a rate of 30 to 60

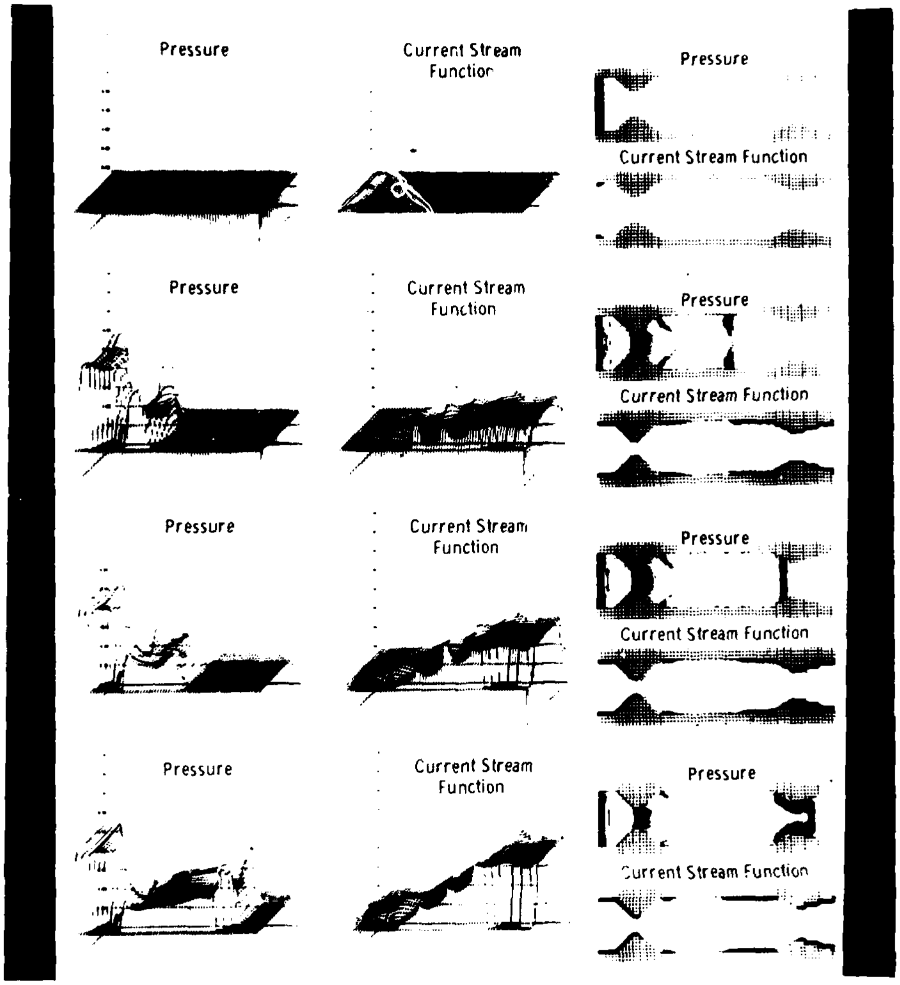


Figure 30. Plasma flow. Magnetohydrodynamic coupling between gas dynamic and electromagnetic force fields - in animation.

Figure 31. F-16 in a roll. Animation.

pictures/s of real time. In other words, the frame buffer containing the picture data can be refreshed with new data at a rate of 30 or 60/s, a task not suitable for most computer graphics systems. The primary failing is that computers do not compute fast enough.

An approach to expediting the calculation is to hardwire the computer number crunching with specially dedicated chips which can perform the work of software: transformations, hidden line/surface processing, color/shade mapping, clipping, etc., at a rate orders of magnitude faster than the traditional software methods. Thus, for economic reasons, real time graphics is primarily accomplished through hardwiring (special computers) and not the conventional host (software) to graphics device method. An example of real time is shown in Figure 32, an aircraft runway modeled to simulate flight conditions when landing*. If stroke or line plotters are used instead of raster scan devices, much fewer data are plotted; and although realism is sacrificed, real time display becomes possible.

---

*Evans and Sutherland

Figure 32. Real time. Simulation of an aircraft landing on a runway. Evans & Sutherland, Salt Lake City, Utah.

VII. FUTURE TRENDS IN COMPUTER GRAPHICS

Hundreds of graphics hardware vendors exist today, and the list keeps growing, along with the list of large mainframe computer companies (IBM, DEC, Hewlett-Packard) that are making graphics devices standard peripherals to their computers. The answer to why computer graphics (especially color) is suddenly coming into vogue may lie in a number of factors:

a. The economics of computer graphics hardware is changing rapidly with recent advances made by the industry in microprocessor technology. The cost today of a typical raster CRT system of 512 x 512 with 256 color intensities is approximately $15,000 to $20,000 including the monitor ($2,000-$5,000), the video controller ($2,000-$4,000), memory ($6,000-$7,000) and color look-up table ($1,000-$3,000). Since memory and monitor costs may drop significantly, the prospect of existing systems selling for half their present cost is a strong possibility.

b. Calculations or simulations have become increasingly sophisticated, paralleling advances made with high-speed computers; and computer graphics has moved from the realm of luxury to that of necessity.

c. Multidimensional color graphics is much more interesting to work with than raw data.

d. Graphics software packages are on the rise and more accessible and comprehendible to the engineer.

e. The field of computer graphics--in engineering, architecture, business etc.--is just in its infancy.

In summary, a number of techniques were presented to visualize multidimensional scientific data: contours, 3-D surface mapping, vectors, trace particles, objects in perspective, etc. The problems encountered in treating voluminous data were: trying to display some semblance of the overall grid while simultaneously discarding irrelevant data; occulting nonvisible surfaces; perceiving depth for 3-D objects; and painting or shading. The problems can all be resolved by judicious coordination of software and hardware capabilities. Realism can play a major role in understanding data not only in 3-D scenes, but also in 3-D surface mapping, and tracer particle and vector plots. Animation is also fundamental to understanding 3-D data, because not only does it provide for a format for exhibiting time-dependent motion, but it also allows for examination of steady state data by sweeping a grid in segments while maintaining an overview of the entire grid during the complete sequence. Kinetic depth perception used in animation is an effective means of resolving the depth problem.

Computer graphics is just now establishing itself as a permanent fixture in the exploitation of computer technology; and its role in the analysis of multidimensional scientific data will increase significantly in the coming years. Just as timesharing and interactive terminals preempted batch data-entry and punched cards, computer graphics will replace most line printing for many engineering applications--and at a cost far less than existing cost.

Most of the illustrations in this report were produced by graphics equipment, each piece costing no more than $10,000 to $18,000. Computer time varied from a few seconds for the airplane, contour-topographical maps, tracer particles, etc., to no more than a minute (CDC 6600 equivalent) for the image processing of the baby's face. A comprehensive software package SCAN small enough to be accommodated by microcomputer performed some of these diverse tasks.

In summary, computer graphics is no longer a luxury for only the most privileged computer users; it is a fast, inexpensive, important step in analyzing extensive engineering data.

REFERENCES

1. Newman, W. H. and Sproull, R. F., Principles of Interactive Computer Graphics, 2nd Ed., McGraw Hill, 1979.

2. Feldman, Laurence A., "A Numerical Scheme for Predicting Transient Shock, Boundary Layer, and Magnetohydrodynamic Phenomenon" Arnold Engineering Development Center, Arnold Air Force Station, Tenn., Dec. 1978.

3. Feldman, Laurence A., "Computer Animation," AFATL-TR-73-174, Air Force Armament Laboratory, Eglin AFB, FL, Aug. 1973.

4. Whitted, Turner, "An Improved Illuminations Model for Shaded Display," Communications of the ACM, Vol. 23, No. 6, June 1980.

GLOSSARY

Algorithm	A method or numerical procedure for accomplishing a mathematical or computer task
Aliasing	A pixel overlapping the edge of a surface causing a loss in surface resolution
Analog Signal	A signal (set) containing an infinite range of values between two limits contained in one domain
Animation	Continuous motion achieved by the rapid display of stills where objects in each frame are displaced in consecutive picture frames (synthetic motion)
Artifact	A picture discrepancy such as aliasing or transformation discrepancies yielding "holes"
Baud Rate	Speed of communicating data over telecommunication lines, BAUD =
Bit	A binary digit, either 0 or 1
Bit Map	A matrix containing the values of each bit stored as elements
Buffer	Region (memory or disc) for the temporary storage of data in transit
Byte	An eight-bit binary unit of data
Cathode-Ray Tube (CRT)	An image-generating vacuum tube where electrons emanating from a gun impact on a phosphorescent screen creating a black and white, half-tone or color picture
Central Processor Unit (CPU)	Computer hardware where operations are performed
Clipping (Cropping)	A procedure for displaying only part of an object which lies in the picture frame and truncating or discarding that part of the object which lies outside the frame margins
Color Look-up Table	A table which converts bit-map z-axis values into CRT intensity values
Computer Aided Design (CAD)	Employing graphics in designing integrated circuitry on computer chips or pc boards, airplanes, cars, apparrel, etc.
Concatenation	The joining together
Decode	To break down
Digital Signal	A signal (set) containing two values within one domain typically encoded to reproduce analog information

Digitize (video)	Convert images to a set of numbers or digits; a device for converting a picture of lines and shaded regions into a matrix of numbers where each element of the matrix represents the intensity of a section of the original picture
Direct Memory Access (DMA)	Transfer of large quantities of data between memory and high speed peripherals autonomous from the CPU
Discretization	A numerical procedure of dividing a region into discrete cells or zones and solving a set of continuous differential equations over these finite cells
Electrostatic Plotter	Plotting by depositing an electrostatic charge on paper to attract a dark toner material. Requires vectors to be converted to a bit-map, but is much faster and less expensive then electro-mechanical plotting devices
Encode	To build up as opposed to decode, to break down
Field	One portion (typically half) of a picture; fields are combined (interlaced) to form frames
Fill	An algorithm for shading a polygon with a prescribed color or grey-level
Film Recorder	A graphics device which records plots or graphs directly to film
Frame	One complete picture; a movie is made of a series of pictures by advancing the frame
Frame Buffer	Bit map memory used by the video controller to generate raster values for one picture or frame
Graphics Terminal	A CRT graphics device which generates line or raster graphics in black and white or color
Grey-Level or Grey-Scale	A degree of darkness measured on some arbitrary scale from, e.g., black (0) to white (1) with intermediate greys between 0 and 1
Half-tone	Systematically assigning pixels a prescribed pattern in order to create a visual effect; primarily used, for example, when a system is limited to eight colors; by spatially mixing 2 or more then the eight original colors occurs
Hardcopy	Paper or film output
Hardware	As opposed to software, computer equipment including the basic components such as the central processor, disk, tape drives, terminals, printers, plotters and other peripherals
Hertz	Cycles per second

Hidden Line Algorithm	A procedure for masking or occulting lines obstructed by other lines in the image plane
Hidden Surface Algorithm	A procedure for masking or occulting surfaces obstructed by other surfaces in the image plane
Host Computer	Principal computer that generates data and transmits them to the graphics device
Hue	The color of a complex light signal irrespective of its saturation or luminance values
Interactive Graphics	Interacting directly with the computer by sending rapid display modification and dynamic changes; the emphasis is on two-way communication between the user and the computer
Image Plane	A 2-D picture plane containing the projected view of a 3-D scene
Image Processor	A raster scan graphics device capable of varying pixel intensity or color which is capable of generating and processing images
Interlaced	A process of combining fields to form a frame, i.e., the first field contains only odd-numbered scan lines, leaving spaces at even-numbered locations; the second field contains only even-numbered scan lines to fill the spaces
Landsat	Pictures taken of the earth from a Landsat satellite
Line Plotter	A plot device which draws lines by connecting points as opposed to a raster device which generates graphs from a matrix of dots
Luminance	The brightness of a complex light signal
Non-interlaced	Scanning consecutive lines when one field = one frame
Monitor	A device, typically based on a CRT, capable of displaying information, similar to a television receiver
Mainframe	A large computer with memory usually greater than 512K bytes, a 32-64 bit processor and a hard disk, very fast central processor
Memory	Addressable bytes in a computer which stores information
Microcomputer	The smallest computer with a memory less than or equal to 64K bytes, an 8-bit processor and a floppy disk
Microprocessor	Very slow central processor

Minicomputer	A medium size computer with memory in the range of 64K to 512K bytes and a floppy or hard disk, moderate speed central processor
Monochromatic	One color (black or white)
Occult	To hide nonvisible surfaces
Perspective	Projection of a three-dimensional picture on a two-dimensional image plane
Pixel	The smallest addressable area, picture element
Raster	The pattern produced by scanning horizontal lines spaced vertically on a screen
Raster Scan	A device which controls the intensity of each dot or pixel in a rectangular matrix covering the screen
Realism	Employing curved surfaces, shading, and hidden surface algorithms to achieve graphics output which imitates real life imagery
Real Time	Clock-on-the-wall time as opposed to computer time or phenomenon time period
Refresh Memory	Video memory storing picture matrix
Resolution	Degree of accuracy of picture determined by the number of dots/inch across the screen
Saturation	The intensity of color compared to white in a complex light signal (i.e., 100% green contains no white light)
Scan Conversion	Converting vectors to rasters
Simulation	Predicting the behavior of a physical system by analogous model
Software	Computer programs written in FORTRAN, PASCAL, etc., which communicate algorithms to the computer
Stand-Alone	Having its own computer capability
Transformations	Translate, rotate and scale
Vector Graphics	Graphics techniques which utilize lines, polygons and polygon fill
Video Controller	Device which generate signals to be sent to the monitor
Virtual Memory	Slow memory that appears to exist to the programmer even though it is not fast addressable memory; techniques of addressinq more memory than is physically present in order to simplify programming at the expense of other computer resources

DATE

ILME