Title
A USER'S GUIDE TO COMPUTER-GENERATED MOVIES

Author
Mahoney, J.

Publication Date
1983

PUB-3030 c.1

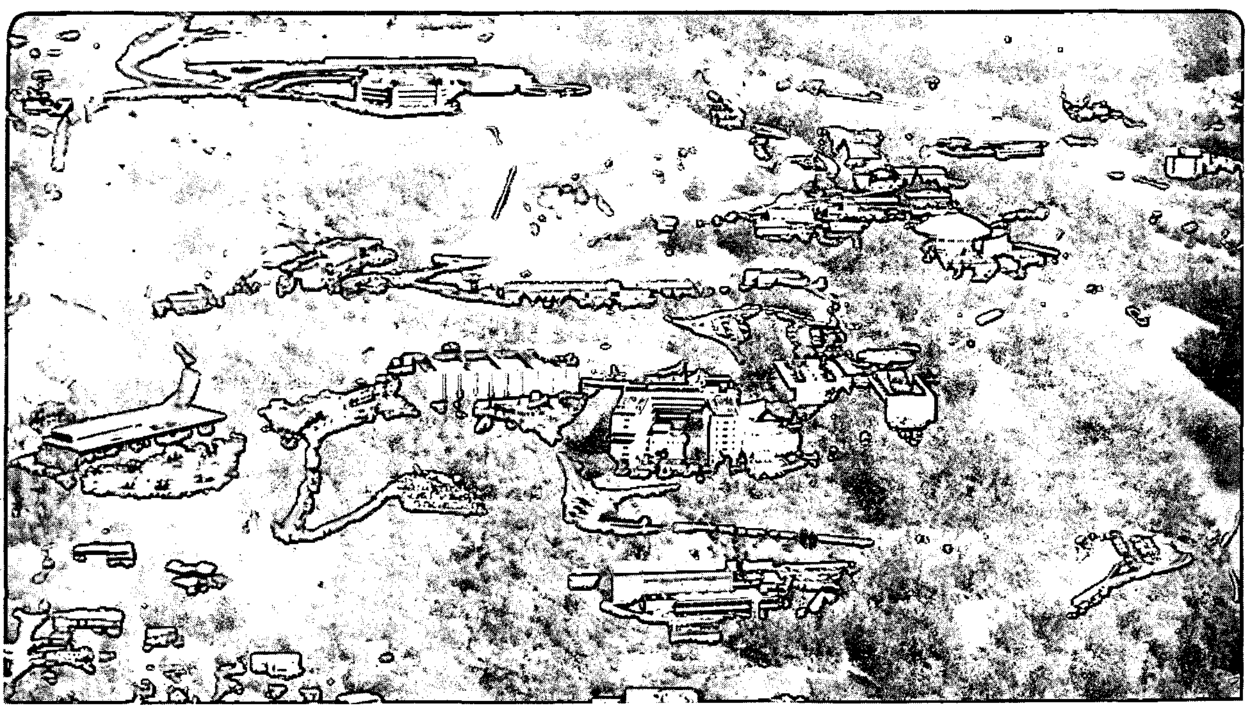
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Engineering & Technical Services Division

A USER'S GUIDE TO COMPUTER-GENERATED MOVIES

Jeannette Mahoney

January 1983



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

PUB-3030

A USER'S GUIDE TO COMPUTER-GENERATED MOVIES

Jeannette Mahoney

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

January 1983

This work was supported by the U.S. Department of Energy under Contract Number DE-AC03-76SF00098.

A Users' Guide to Computer-Generated Movies

Jeannette Mahoney

Acknowledgements

The production of this manual was a joint Lawrence Berkeley Laboratory (LBL) and Lawrence Livermore National Laboratory (LLNL) project. The computer facilities used were those of the National Magnetic Fusion Energy Computer Center (NMFECC), which is located just outside LLNL and uses the LLNL film recorders and optical laboratory. Most of the programming was done in Berkeley, through a remote user facility of the LBL Magnetic Fusion Energy Department. The author's hands-on experience is limited to the programming only; the film recording and optical processing are done inside the Livermore National Laboratory.

The idea for the manual was conceived by Don Vickers of LLNL, under whose guidance and tutelage it was begun. The author is indebted to him and to Carolyn Hunt and John Blunden, also of LLNL, for much helpful advice and criticism, and to Paul Renard of LLNL and Rick Steele and Eliezer Rosengaus of LBL for their help in the use of the respective computer equipment and routines.

CONTENTS

A Users' Guide to Computer-Generated Movies

Jeannette Mahoney

INTRODUCTION

Computer-generated movies have been used for many years in commercial applications, for educational presentations, and as an experimental art medium. Hardware and software innovations, however, have now moved them out of the realm of esoterica and by reducing the cost and increasing the accessibility of movie making routines, made them available for general use.

Movies can aid in interpretation of physical shapes and processes by adding depth simulation, or by compressing time to compare various steps of a long term process. Conversely, time can be lengthened to investigate movements too rapid to be seen under normal circumstances.

Sound and color can be added to draw attention to a particular stage of a process, and also to distract the viewer from minor technical problems in the visual portion. They can also be used to represent variables.

One very important application of the computer movie is in data analysis. Engineers have been using programs designed to simplify and present calculations on complicated problems for some time now, and more recently experimentalists in the physical and life sciences have begun to work with their data in a similar way.

The expense involved in making movies is obviously a serious consideration and if prohibitive overrides any advantages of ease and accessibility in creating them. Consequently, the cost of using different algorithms and packages as well as the various techniques described in this manual will be discussed and compared with alternative methods. Movies are always expensive to make, but there are techniques and tricks that help keep the cost to a minimum.

Most importantly, this manual is written not for the computer professional, but for the Fortran programmer who uses a computer only as a tool to analyze data. It is hoped that the manual will enhance the usefulness of that tool.

PART I: TECHNIQUES, TRICKS OF THE TRADE

Types of Computer Movies

Computer-generated movies range from fairly simple processes to quite complicated procedures that result in a sophisticated product. In general there are two types, instructional and analytical. Analytical movies are usually simple, often omitting both color and sound (a movie without sound is called "footage"), while demonstrational movies are sometimes quite elaborate and make use of the latest techniques available, usually becoming rather expensive in the process.

An excellent example of the instructional genre is one movie that gives an animated representation of reduction of orthogonal matrices, using spheres to represent the matrix elements as a voice explains the mathematical procedure.

One which is both demonstrational and analytical shows a map of California on which points are lit up sequentially to represent earthquakes that occurred in that area in the years that appear simultaneously above the map. In the earlier years, for which data on only major earthquakes are available, the lights are sparse and go on and off slowly; in the more recent years, when earthquakes of even very small magnitude were recorded, the map twinkles like a Christmas tree. Even so, clusters, rather than a random distribution, are very prominent along the fault lines.

In the realm of analytical movies, there are many with engineering applications. For example, a simulation has been made of gases exploding in the cylinder of an internal combustion engine. Colored marker particles are used to show the movement and distribution of the gas. An interesting result of making this movie was that when it was viewed, sonic vibrations were seen to be reflected back into the gas, an effect that is certainly understood but that had not been anticipated. This sort of animated display often brings out effects that were unexpected by the engineers or scientists making the calculations.

Computers have also been used to do finite element calculations on complex structures such as bridges, then to create animated displays of the stresses and strains on each section of the structure. Engineers have found this sort of technique extremely useful in design. Another use of animation in engineering design is one in which an interactive program is given the pivot points for a machine and then creates an animated display of a device that will suit the requirements.

GENERAL

The actual sequence of creating a movie is straightforward. The data is fed into the large computer where all the complicated calculations take place. Graphics and animation packages, called viewing algorithms, can be used for all or part of these calculations, or the user can create her or his own Fortran program and load it with one of the available graphics libraries. A picture can then be output to a TV screen, modified interactively until the desired result is obtained, and then sent to a film recorder (and/or to a printer if a hard copy is desired), where minor calculations can first be done on a minicomputer after which it is photographed onto one frame of a movie film. When all the frames have been recorded, the film is taken to the optical lab for the final processing. Often, short film strips are made and edited, then combined when a copy is made at the optical lab. A copy should always be made for projection and the original saved.

Perspective, "Depth Cues"

Computing Perspective

Perspective is simulated in two dimensions on a screen much the same way an artist does it on paper, by assuming a convergence point well behind the viewing plane. With computer graphics, however, the entire object must be envisioned in perspective behind the plane of the screen, with its two dimensional projection appearing on the screen. When computing this projection, one must consider the position of the viewer, the distance of the viewer from the screen and the field of view. Most projections are made for a 90 degree field, which would put the viewer at the face of the screen. When the viewer is 2 or 3 feet away the field of view is only about 10 degrees, so unless the correct field is specified objects will appear distorted when seen from the ordinary viewing distance. The coordinate convention is important also, since some available algorithms which use a right-handed coordinate system are typically used in conjunction with others that use a left handed system, so the appropriate conversions must be made to keep these routines mutually compatible.

Time-Resolved Parallax

One advantage of using motion to create a three dimensional effect is that the viewer retains the memory of an image after it has changed; showing different positions of an object in sequence gives the illusion of three dimensions with the viewer's imagination supplying the depth. This phenomenon has been exploited to particular advantage in a film of a rotating DNA molecule, in which the double helical structure can be seen very clearly during the rotation although it is difficult to envision from a still.

Intensity, Superposition, Texture, Shading and Highlights

Another depth simulator is the appearance of distance caused by diminished intensity. Hidden line and surface algorithms compute what lines or surfaces would be obscured by other surfaces appearing in front of them and remove them from the picture. Facets of a surface can be colored or shaded differently to appear to lie in different planes from the source of light. Texture can be simulated, then colored and shaded to cause a surface to appear farther from the viewer. Highlights appearing on a convex surface and moving as the surface is rotated produce a very realistic depth effect. These can be generated by an LLNL routine for which the user supplies only the positions of the object and the light source. Both shading and highlighting can be done off line on a minicomputer, if one is prepared to write the algorithms for them. The existing routines run on the mainframe computers only.

Shading and Silhouette Smoothing

The term "shading" refers to a group of techniques designed to alter the appearance of the surface being displayed; these techniques do not necessarily shade in the normal sense of the word. Surface patches, described on the next page, for example, are considered shading techniques.

Faceted or Flat Shading

Solid objects calculated by several viewing algorithms consist of polygons joined together to create the desired shape. The polygonal structure of the surfaces formed can be emphasized by faceted or flat shading, where each facet of the surface is a constant shade over its entire area, although individual facets may differ in shade from each other. This type of shading makes the line formed by the meeting of two polygons more prominant and gives the appearance of two planes meeting at a sharp junction.

Smooth Shading

Used to help eliminate the lines between polygons, the shade is defined at each end of the CRT scan line, or at the beginning with a shade difference rate specified, and is varied smoothly along the scan line. This type of shading deemphasizes the junction and gives a curved rather than angular appearance to the surface.

Silhouette Smoothing

Another technique to reduce the angular appearance involves rounding off the silhouette. Called silhouette smoothing, it is accomplished by dividing the edge of the object into many smaller polygons to minimize the angular shape of the outline.

Diffuse shading

Diffuse shading is a technique which produces a matte, rather than a glossy, finish.

Fractal shading

Fractal shading is a technique in which the surface is broken up to create a texture. It involves a fairly large amount of calculation and is a very expensive algorithm to run.

Techniques to Reduce Computing Time

Calculations done on any of the large computers can become very expensive if long; the time per frame is kept to a minimum by several tactics.

Superposition of Titles

Titles should be drawn once and superimposed on other frames later at the optical printer, rather than being drawn by the computer for each frame.

Forward Differences

The use of forward differences to calculate distance along a spherical surface is a method of solving quadratic polynomials without using multiplication.

Keyframe animation

Keyframe Animation is another technique used to reduce the computing time on the large computer by having it calculate only a few pictures, called keyframes, and using a minicomputer off line to interpolate between them and generate the intervening frames. If part of the image is background and does not change from frame to frame, it can be generated once, then removed from the data frames and be superimposed later with the optical printer. This technique saves calculation, data transfer and plotting time.

Surface Patching

Complicated surfaces can take up to 15 minutes per frame to calculate, an enormously expensive period of time. To avoid this, smooth surface models are used in the stills and complicated patches superimposed on them. A patch calculated by a cubic polynomial in two variables is called a bi-cubic patch, a quadratic polynomial, a quadratic patch.

GRAPHICS

Vector vs Raster

Printers and cathode ray tubes (CRT's) operate on either a vector or raster principle. Vector devices use analog drivers to draw a line across the screen or page. A vector CRT is limited in number and length of lines; if these quantities exceed the refresh capability of the device, flicker results. A raster is a rectangular matrix of points on a CRT which correspond to locations in memory. A display program scans that block of memory, usually from top to bottom, scanning each row left to right, and if a bit is set, the corresponding point on the screen is displayed. Resolution increases with the number of points in a raster, so as memory becomes cheaper, higher resolution is increasingly available with raster CRT's. Most graphics algorithms are written with vector commands which are converted to the appropriate instructions for raster devices.

Aliasing

Aliasing, also known as "jaggies", occurs on a low resolution raster device when a diagonal line is drawn. Because of the inability of the device to draw a point between two fixed locations, a staircase effect is produced when the line is short relative to the distance between the points. Anti-aliasing is the use of intensity or grouping to offset this effect. Where possible, the image is intensified when the point is centered on the line and reduced when it is off center. Algorithms are also used to redistribute the pattern of the points falling on each side of the line in such a way as to minimize the staircase effect.

Clipping

Whenever an object is drawn, viewing limits must be set, outside of which no part of the object will appear. Removing those parts that lie outside the limits is called clipping. Each time the object is rotated or translated it must be clipped again. A fairly simple calculation, the clipping algorithm consists of checking each point or line against the four sides of a rectangular viewing area. This algorithm is available in all of the graphics routines.

Aspect Ratio

The ratio of height to width of movie film is .75. Called the aspect ratio, it is well to keep in mind, since the final picture will be clipped to this ratio as it is being filmed.

ANIMATION

Animation enhances a graphical representation by introducing the dimension of time. For example, three variables can be shown as axes of a three dimensional coordinate system in which the data points are plotted and projected onto a two-dimensional surface. Plotting the points with different colors can add another parameter to the representation. A true three dimensional effect cannot usually be achieved, however, without movement. Animation creates the illusion of three dimensions by time-resolved parallax, the phenomenon caused by the viewer's remembering the last position and supplying the perspective simply through imagination. All the graphics packages and libraries contain routines for translation, scaling, rotation and reflection of images.

Timing

Timing for animation is both critical and difficult. Two aspects which must be considered are that the movement be consistent with what the eye can follow and that action be presented at a rate comfortable for the mind to follow. Too rapid movement can result in double imaging and/or motion blur, look unnatural, or not give the viewer time to assimilate what was displayed. Too slow movement can become boring. Getting the correct timing is most often accomplished by trial and error, although there are some guidelines. To avoid double imaging in translation of an object, the distance moved must be less than about .5Δ of the image width. The time required to traverse the screen will be 8 or 9 seconds. In rotation, a full-sized image should not exceed 1 degree of rotation per frame, although smaller images can rotate more rapidly.

Usually each frame is duplicated at the optical printer so that the finished movie contains two identical frames for each picture. This technique, called double-framing, slows down the action somewhat and minimizes jerkiness. Two frames per picture is optimal; three will start to create jerkiness again.

Acceleration

Acceleration is also an important factor. Movement should not begin and end abruptly or proceed with a constant velocity. Variations in speed depend on the effect intended, of course, but in general a stationary object should start to move slowly, reach a constant velocity, then slow down for several frames before coming to rest. In both translation and rotation, it should take about 2 seconds to reach a constant velocity. This sort of variation is especially important between successive sequences and film strips.

Two objects can be rotated at different speeds to emphasize an action or to keep the two objects distinct from each other during a sequence. In an animated simulation of a chemical reaction two molecules were simultaneously rotated and translated at different speeds to show how one fits inside the other as it enters and forms a chemical bond with it. The sequence is very effective because the rotating molecules give a spacial framework for the model to show how the reaction can take place.

Sound

Using sound it is possible to represent two more parameters of data by making one proportional to frequency and the other to volume and changing the appropriate sound constituent as the value of the parameter changes. Sound can be drawn in as a wave form at the side of a picture either with a graphics routine or by a CRT input device, such as a light pen or a graphics tablet. Sound track film has sprockets on one side only, the other edge being the sound track. Ordinarily, a picture is projected onto a small rectangle inside the larger image area that contains the sound track. By projecting the picture over the whole area, one can simultaneously create the sound track from the computer-generated picture. If the individual frames of the film strip are not exactly aligned, however, the film must be cut and spliced diagonally, as with audio tape, to avoid hissing or popping sounds between frames.

Keyframing

Keyframing, drawing only a few keyframes and interpolating between them, is useful with line drawings to save computing time. This technique can be used to supply the intervening frames for different positions of an object or to transform one image into another. Even complicated patterns can be changed with merely a linear interpolation; it is only necessary to start out with the same number of segments for the original and final images and to specify the number of steps or frames desired between the two. Keyframing can also be done for fadeouts at the optical lab by creating the beginning and final intensities with computer pictures and having the optical printer make the intervening frames. As mentioned earlier, an unchanging background can be created once as a keyframe and combined with the changing foreground frames by the optical printer.

Technical and Mechanical Considerations

Film

For one-time use, the movie can be shot directly onto 16 mm film and the original displayed, but for a film that might be used again, the original should never be displayed. The film can be damaged in either transporting or projecting, so an optical master should be made for projection and the original saved. For best resolution, 35 mm film should be used for the exposure and optically reduced to 16 mm.

Projection

Variations in the viewing area, lighting, projector bulbs and the distance from the projector can all affect the finished product. It is well to remember that only at the projector is the perspective optimized, and to consider the size of the projection room when making the pictures.

Intensity

Variations in intensity on the film are inevitable. A common technique to compensate for differences in film response is to make an identical grey patch on the first frame of each run and to match all the runs at the optical lab.

Film vs Video

When making pictures for video tape, care must be taken not to run off the edge, since the viewing area on video tape is smaller than that of film.

Frame Numbering

Each frame should be numbered, so that if a correction is needed after filming, the frame can be readily identified. The numbers can be masked out at the optical lab when the copy is made.

COLOR

Color adds an entirely new dimension to a movie; it can be used to emphasize certain parts of a sequence, to help distinguish between surfaces or objects, for its aesthetic effect, or even to represent a variable that changes with time. It is also a very tricky medium to work with and to use properly.

There are many sophisticated techniques applied to the use of color, some of which are contained in viewing algorithms available from the library. One which can be run off line computes highlights that would appear on a stationary or rotating surface from a fixed light source and also contains algorithms for smooth and diffuse shading. Most of the graphics libraries and applications programs contain similar shading algorithms.

Color Filters

The filters used in film recorders each filter out all wave lengths of light except the one desired (e.g., a blue filter allows only blue light through), and since different wave lengths have different focal lengths, a color compensating lens is used to focus all the colors at the same point. An RBG (red blue green) system consists of the three primary color filters which can be added together in sequential passes on the same frame to produce the secondaries and is called an additive system. Systems composed of secondary-color filters need two filters for each color and are called subtractive. Additive systems have the advantage of using only one filter at a time, thus minimizing reduction in light intensity.

Variations in Response

Variations are inevitable in color film processing: differences in temperature, in the composition of the chemicals, in the film itself, and some say in the phase of the moon all affect the results, making it difficult to maintain consistency over a long period of time. Consequently, the closer together all the frames are exposed and processed, the higher the likelihood that the colors will not vary much from frame to frame.

Choice of Colors

Care must be taken in the choice of the colors themselves. The three primary colors used in color photography are red, green and blue. The secondaries, formed by pairs of the primaries, are yellow, cyan and magenta. Since all the primaries together form white, and since each secondary is formed by two primaries, a combination of any two secondaries will also result in white. The intersection of two colored lines will turn white or become a different color unless the colors are chosen carefully, i.e., a primary and a secondary, one of which contains the other. The intersection of a red line with a yellow line will be yellow, which is all right, but so will the intersection of a red line with a green line, which is not all right.

Bleeding

When colors abut each other the same mixing of color can occur, caused by "bleeding", or "blooming", the phenomenon of scattered light exposing adjacent areas on the film. The result can be color bands or white lines from two colors combining or simply a blurred area. This can be prevented by leaving black spaces between the areas which prevent bleeding and are not easily seen in the projected film.

Mach Bands

Often when bands of color abut each other an interesting optical effect occurs which gives the illusion of a lessening of intensity near the edges, resulting in a "Mach band" of deeper intensity appearing at the center. Shading algorithms can compensate for this effect.

Overuse of Color

Using colors that are too bright can result in overexposure, giving exactly the opposite effect of that intended. Overuse of color should be avoided also, particularly in labeling - labels should be white or of only one color to prevent the picture from becoming confusing.

OPTICAL PRINTING

The final stage in making a movie is having the film processed at the optical lab. It is important to stress that any film which will be shown more that once should be saved and a copy made for projection because the film is likely to be scratched or otherwise damaged by the projector and in handling. For high resolution the original film should be 35 mm. The optical printer can reduce it to 16 mm if necessary when the copy is being made. The 16 mm copy then becomes the optical master, while the 35 mm original is saved for future copies. For presentation movies it is also advisable to save the output file.

During the optical printing special effects can be added. There are also many routine jobs that are done optically more easily than with the computer, most of which have been mentioned already. Title frames are duplicated and titles superimposed here; frame numbers are masked out and each frame duplicated at this point; fadeouts or dissolves should be done at this time.

Color can be supplied if black and white film has been used, although it is often preferable to do the original filming in color. Compensation can be made for intensity variations by matching gray patches on the originals with each other on the copies. Although some algorithms are available to create transparency on the original picture, it can also be simulated at the optical lab.

Many printer techniques are preferable to computer algorithms because of the very low cost involved in optical printing compared with computer time. For a long optical effect which can be controlled automatically, the per-frame cost on the optical printer is negligible, but could be quite high on a main-frame computer.

Copying and Reduction

The printer can make one to one positive or negative prints, reduce 35 mm to 16, provide a spliceless print from several originals, or reprint several cycles of a repeated motion.

Repair and Improvements

The printer can remove surface scratches from an original using a "liquid gate". It can also compensate for changes in exposure, color balance, size, position and tilt which may be caused by recorder or processing drift between runs. Also, action can be speeded up, slowed down or reversed.

Matching

A computer simulation can be matched in size, position, timing or intensity to live photography or to another simulation.

Masking

A matte, or high contrast mask, can be used to superimpose titles or to mask out portions of pictures such as frame numbers. The masking technique is also used to add background or foreground to a picture and to combine several images in a montage.

Color

Color can be produced from black and white separations, and masking techniques can also be used to prevent color bleeding and mixing.

Dissolves

The printer can provide a fade out or a fade in, as well as a dissolve between two scenes. A technique known as "cross dissolve" is used to minimize jerkiness when only a few frames are available, such as in satellite weather pictures.

Expanding

A few frames of computer output can be expanded into a longer film by many techniques in addition to double framing, such as freeze framing, changes in speed, reverse motion, repositioning, magnifying part of the image and combining several different images. When computer cost per frame is high, post-recording manipulation such as this is an attractive alternative.

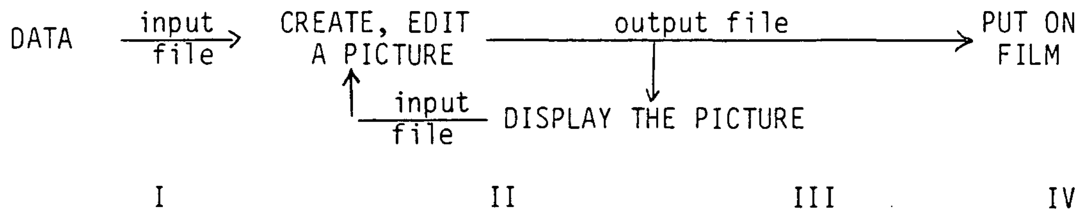
COST COMPARED WITH COMMERCIAL MOVIEMAKERS

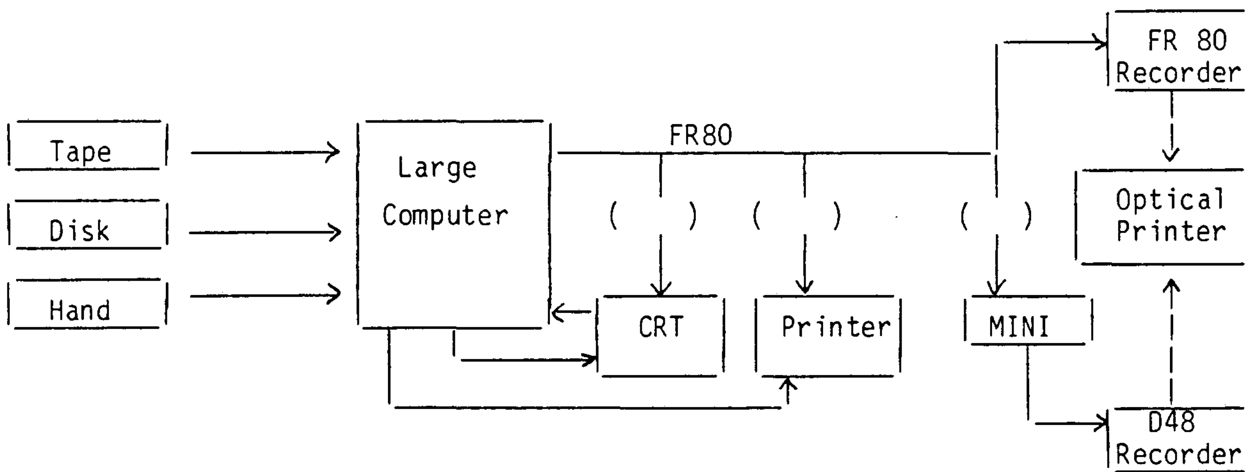
A movie produced commercially will cost from $1000 to $3000 per second of viewing time, depending on how many special effects and latest techniques are used. The cost of a movie produced at Livermore is determined by the computer time used, since unless very long raster plotting time is necessary, the film recorder use, film cost and personnel time is included in the I/O overhead and not charged directly.

PART II - MAKING A MOVIE

Anything that can draw a picture can make a movie.

Sequence:





The schematic shown above depicts a typical sequence for creating frames of film. The boxes on the right contain the output devices used, which can be accessed either directly or from the FR80 file through conversion routines represented by the path breaks in parentheses. The FR80 file goes directly to the FR80 film recorder (see output devices) and is currently the standard graphics output file. FR80 files can also be sent to the D48 by changing one letter of the file name. The broken line indicates a path that is not automatic, i.e., film must be hand-carried to the photo lab. Because of the expense involved in running on the film recorders, it is advisable to send all output to TV or printer devices until a satisfactory final version is created, and then and only then, start putting the pictures out to be filmed.

The algorithms for creating and editing pictures are contained in graphics libraries and in packages, as are the routines for rotating and translating them. The routines for reading the data, drawing and displaying the images and outputting to the appropriate file can all be contained in a user-written Fortran program, and will vary depending on which graphics library is used. There are also general application graphics programs for which the user need only enter the data. A library called GRAFLIB is supported by the LLNL computations group and is machine, device and compiler independent. It is also available at NMFECC.

Appendix I contains an partial description of the NMFECC System.

Output Devices

RJET

An RJET, remote job entry terminal, is a station controlled by a mini-computer which contains I/O devices such as printers, plotters, graphics terminals, etc. Each RJET consists of the devices needed at its particular location. A similar station for off-site users is called a RUSS.

Film Recorders

Movies generated by computers are simply a series of stills produced by a graphics routine and sent to an output file which is then displayed on a CRT screen and photographed by a camera. The device that displays and photographs the CRT screen is called a film recorder. Vector devices produce film at the rate of ~1 second per frame; raster devices take longer, depending on the resolution and whether color is used. There are two at LLNL, which are used by the NMFECC - the FR80 and the Dicomed D48. Because the FR80 was the first, and for a time the only, film recorder, the computer system has an FR80 file that is used for both recorders, with a conversion routine translating the format for the D48. Device-independent files can also be made, then converted to the appropriate format. Both devices use white phosphor CRT displays with color filters.

Information International Film Recorder (FR80)

Input for the FR80 can be directly from the FR80 output file. At present, the FR80 is not reliable for color and is mainly employed in producing black and white microfiche. It is a vector device and has a resolution of 16K by 16K. There are 9 film types available, chosen by specifying the appropriate camera number, as follows:

1	105 mm fiche (48x reduction)
2	35 mm
3	16 mm sprocketed (perforated)
4	16 mm unsprocketed (unperforated)
5	105 mm fiche (24x reduction) conventional format
6	105 mm (24x reduction) document format
7	12 inch black on white film for hardcopy
8	35 mm color
9	16 mm color

Dicomed D48

The D48 has both raster and vector capability, with vector addressability of 32K by 32K and raster resolution of 4K by 4K. A VARIAN V75 minicomputer, which drives the D48 can accept FR80 data format (vector), two raster formats, and the D48 native code which can be vector or raster or both. The VARIAN also controls the filter operation, advances the film in the camera and can run algorithms to do shading and highlighting. A black and white CRT is used because a white light image is of much higher quality than a color image and the light intensity is constant. The recorder has an RBG (red, blue, green) filter system, containing 7 filters, one of which is always in place. The D48 has very high-resolution color capability and consequently is the preferred (and at this time the only) device for color pictures. It produces 35 mm movies in color and black and white, 35 mm color slides, 16 mm color movies, and 4X5 sheet for color and black and white prints, color and black and white negatives, and color positive transparencies.

Dicomed output can be specified in a Fortran program using Graflib; if Graflib is not being used, FR80 output files can be created from the Fortran program and the file names changed later to route them to the appropriate Dicomed film specification. Examples are shown at the end of the following outline.

Outline for Making a Movie

I Input Data

Data can be input from a disk, a magnetic tape, or by hand from the terminal.

The general library routine, LINK, can be used to connect the input and output files to the proper devices. It also creates a dropfile needed for graphics output. The call to LINK specifying input from a terminal for the CRAY is:

```
call LINK("unit59=terminal//")
```

If data is to be entered from disk or tape, the appropriate input files must also be specified.

II Create, Display, Edit a Picture

Graphics routines may be called from any of the three graphics libraries currently available on the NMFECC system. Graflib, the in-house library of the LLNL computer center, contains output device initialization routines in addition to graphics. Disspla is a commercial library produced by Integrated Software Systems Corporation (ISSCO) which offers very high level graphics subroutines. Since it requires a large amount of memory, if a large amount of data is involved it might be advisable to run on a Cray. Both support a wide range of graphics capability and both must be loaded in addition to the standard Fortran library, Fortlib. Graphics and device initialization routines are also available from TV80lib, which is part of Fortlib and consequently is loaded automatically. When using DISSPLA for the graphics, output device initializations are done through TV80lib.

The general format when using either Graflib or Disspla is to initialize the library, draw and display the pictures, then end the library when finished. Graflib may be initialized in two ways, direct and picture file. If picture file is used, then the picture may be composed of several segments, which are defined separately and may be changed independently of each other.

At present, the three-dimensional plotting routines in Graflib are in the process of being documented, so the library Disspla was used to create the graphics for the examples. A reference manual and a users' manual contain instructions for use of the Disspla routines. Brief descriptions of DISSPLA and examples of graphics and of a short movie follow this outline.

All the routines described were run on the Cray.

III Send to Appropriate Output Files

To initialize the output device files, an ID routine must be called for each file. Unless Graflib has been initialized, these routines will be called from TV80lib as described here. To view output on a Tektronix terminal:

```
call TEKID(baud)
```

A call to FR80ID will output commands for each picture to a file or set of files to be displayed and photographed later onto a frame of film either in the FR80 or the D48 film recorder. The names of all FR80 files begin with the letter f and are labeled and sequentially numbered in a format which varies with the machine used. The file(s) may, and should in the case of a movie that may be filmed again, be saved and stored temporarily in the user's computer space by a call to KEEP80 made before the call to FR80ID. It, or they, can then be filed permanently in FILEM (see utility routines in Appendix I) before being sent to a film recorder.

```
call KEEP80(1,3)
call FR80ID(ID,security status,camera no.,no. of blank frames)
```

ID is an 8-character (10 on the CDC 7600) identification which will appear on the first and last frames of the FR80 file.

The security status code in the example on p. 20 is for unclassified. The camera number codes are listed on p. 13.

The number of blank frames desired between pictures is entered as the last argument (integer).

While editing a picture, it is advisable to either insert a c before the FR80ID call or leave it out until you are ready to create the file; otherwise, you may end up with several sets of FR80 files and not know which is the good one when you are ready to send it to the film recorder. Conversely, removing the TEKID call before creating the FR80 file saves plotting time by eliminating the display.

IV Put on Film

To send the FR80 file to the FR80 film recorder, use the GIVE utility routine. If the file is to be stored permanently, write it to FILEM (see App. I) before using GIVE, since GIVE removes the file from the workspace.

GIVE FILENAME (GIVE F* will send all files beginning with f)

will send FILENAME to the FR80 film recorder. For a first try, it is all right to make 16 mm output directly on the FR80. The film from the FR80, however, is extremely brittle and must be handled carefully. A short film should be viewed on a film editor rather than a projector. Later, a file specifying 35 mm film for higher resolution can be created, transferred to magnetic tape and sent to the Dicomed. There it can then be filmed in color, if desired, and the film reduced to 16 mm at the photo lab, where the copying, titles and special effects will be done. In either case, the original file should be saved and stored.

Summary

MOVIE 1

Fortran Program

I Input Data: call LINK

II Drawing, Editing: call TEKID
call library initialization and drawing routines and create a few pictures.

Run program, viewing pictures on screen; edit program; rerun until pictures are satisfactory. Number each picture.*

III Make files for film recorder: call KEEP80
call FR80ID
(replace call TEKID)

Run program with the full number of pictures to create the output files, then transfer them with the following utility routines.

Utility Programs

IV Put on Film

To save the file: FILEM write FR80FILE1 FR80FILE2 .....

To send to FR80: GIVE FR80FILE1 FR80FILE2 .....
or
GIVE F*

To send to D48 If DISSPLA used, create FR80 files as in step III, then change the file names, using utility routine SWITCH, (See App. I) as follows, (abcde are random characters, xx are two digits, sequentially numbered):

35-mm color:	FR80 only	D48 only
CRAY	FFabcdxx	FDabcdxx
7600	FXK35abcde	FXD35abcde
16-mm color:		
CRAY	FGabcdxx	FEabcdxx
7600	FXK16abcde	FXD16abcde

and use GIVE routine as above.

*Often a program bug can affect a few frames only; if they are numbered those frames can be corrected without recalculating the whole series, a process which could be time-consuming and expensive.

DISSPLA

In Fortran Program, "NAME":

All argument names follow the usual Fortran convention for real and integer. Real arguments can be passed without a decimal point when they are whole numbers.

1. Call LINK and initialize devices as described on page 14.

2. Initialize library:

   Call PLTS

3. Begin plot:

   Call BGNPL(i) i=plot no. If i negative, summary on plot only; positive, summary on printout also; equal to 0, no summary.

4. Define coordinates:

   Call GRAF(xmin,xstep,xmax,ymin,ystep,ymax) user units

5. Plot curve:

   Call CURVE(xarray,yarray,npts,marker code)

6. Display:

   Call DENDPL(i) DENDPL ends and displays the plot.

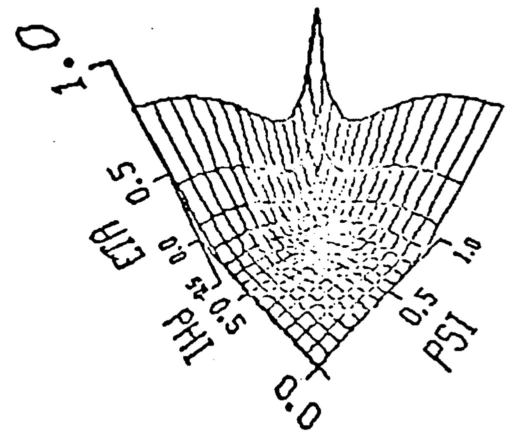
7. End Library:

   Call DONEPL

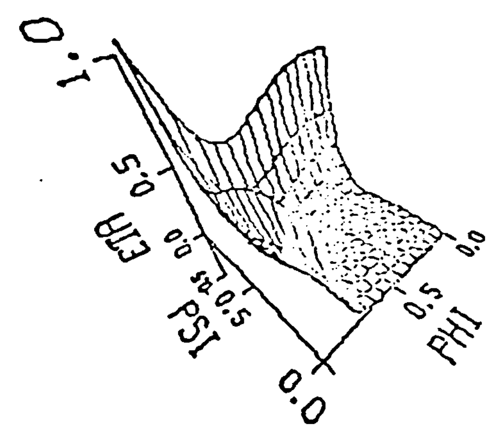
Compile, load, execute: RCFT i=NAME,lib=DISSPLA,go

Typical 2-d plotting sequence:

```
call plts
call bgnpl(-1)
call graf(0,2.,10.,0,50,300)
call curve(xarray,yarray,500,1)
call dendpl(-1)
call donepl
```



Φ - 225, Θ - 45



Φ - 135, Θ - 45

```
c  example of disspla using surfun
c
      external zfunc
      dimension iwork(300)
      call link("unit59=terminal,unit9=terminal//")
      call tekid(2400)
      call keep80(1,3)
      call fr80id("box b50$",0,2,1)
      call plts
      theta = 45
      radius = +1.5
    1 x0=3
      y0=.2
      do 100 i=135,225,90
      phi = i
      call bgnpl(0)
c
c   set alphabets
      call basalf("standard")
      call mixalf("greek")
c
c   set origins for subplots and suppress border
      call physor(x0,y0)
      call nobrdr
c
c   define subplot areas and axes and set viewing angles
      call titl3d("",0,3.,3.)
      call vuangl(phi,theta,radius)
      call axes3d("psi$",100,"phi$",100,"eta$",100,0.,0.,0.)
      call zlevel(0)
      call graf3d(0.,"scale",1.,0,"scale",1.,-.1,"scale",1.)

    labels for subplots
      num = phi
      ithet=theta
      call messag("(f) = $",100,.5,0)
      call intno(num,"abut","abut")
      call messag(", (q) = $",100,"abut","abut")
      call intno(ithet,"abut","abut")
c
c   draw surface
      call surfun(zfunc,4,.07,4,.07,iwork)
c
c   define as one of multiple plots on the same page
      call endgr
      y0=y0+6.
  100 continue
c
c  end picture and display
      call dendpl(0)
      call donepl
      end
c
c   function for z
      function zfunc(x,y)
      alpha=3.6652
      beta=2.8884
      part1=(3./5.)*(x**5+y**5)
      part2=(alpha/2.)*(x**6*(1.-(6.*x**2)/5.)+y**6*
     1(1.-(6.*y**2)/5.))
      part3=beta*x**3*y**3*(1.-(3./5.)*(x**2+y**2))
      zfunc=(part1-part2-part3)/(x**3+y**3)
      return
      end
```

MOVIE 1

Short Movie made using DISSPLA. The circled lines were entered by the user; all others were from the computer. The source code was listed from the text editor TEDI.

```
t1,66
c   movie1: animation of bills data
c
      call link("unit59=terminal,unit9=terminal//")
c     call tekid(2400)
      call keep80(1,3)
      call fr80id("box b50$",1,3,0)
      call plts
c
c   rotate in phi with theta=0
      j=1
      theta = 0
      radius = +3
    1 x0=0
      y0=0
      k=0
      do 50 i=120,225
      phi = i
      call plot(phi,theta,radius,x0,y0)
   50 continue
c
c   rotate in theta with phi=225
      phi=225.
      theta=0.
      do 100 i=1,50
      theta =i
      call plot(phi,theta,radius,x0,y0)
  100 continue
      call donepl
      end
c
c   plot graphs
c
      subroutine plot(phi,theta,radius,x0,y0)
      external zfunc
      dimension iwork(3000)
      j=j+1
      call bgnpl(-j)
c
c   set origins for subplots
      call nobrdr
      call physor(x0,y0)
      call page(11.,8.5)
      call titl3d(" ",-1,6.,6.)
      call intno(j,0,0)
c
c   set viewing angles
      call vuangl(phi,theta,radius)
      call axes3d("psi$",0,"phi$",0,"eta$",0,1.,1.,2.)
      call zlevel(0)
      call graf3d(.3,.1,.9,.3,.1,.9,-.2,.1,.4)
      call surfun(zfunc,4,.03,4,.03,iwork)
      call dendpl(-j)
      end
c
c   function for z
c
      function zfunc(x,y)
      alpha=3.6652
      beta=2.8884
      part1=(3./5.)*(x**5+y**5)
      part2=(alpha/2.)*(x**6*(1.-(6.*x**2)/5.)+y**6
     1(1.-(6.*y**2)/5.))
      part3=beta*x**3*y**3*(1.-(3./5.)*(x**2+y**2))
      zfunc=(part1-part2-part3)/(x**3+y**3)
      return
      end
*end
movie1   66 lines

 all done
rcft i=movie1,lib=disspla,go / 1 2
CF000 - CFT VERSION -   04/08/82 109m
CF001 - COMPILE TIME =     0.0419 SECONDS
CF002 -        66 LINES,        48 STATEMENTS
controllee name is xmovie1   ,load length= 0133160
%s
RUN +xmoviea
%be
0006.602

 all done
%bc
0005.871
files
      1000 rw   %ated00
       235 re   %tedbak
      1150 rw   bmovie1
        54 rw   disout
    135352 r    f6efnv00
    135465 r    f6efnv01
    135371 r    f6efnv02
    135352 r    f6efnv03
    135504 r    f6efnv04
    120214 r    f6efnv0x
      1502 rw   lmovie1
     16165 rw   mmovie1
       235 r    movie1
    215757 re   xmovie1

 all done
give f*

 all done
```

Routines used in Examples

To set up subplot area, dimension and label axes:

TITL3D - gives title for plot and physical dimensions for x and y axes. Dimensions are in inches by default; they can be changed by calling UNITS (scale) where scale is "inch" or "in", "cent" or "cm", "mill" or "mm", or any real scaling factor.

The title is entered as two arguments: a text string in "" and an integer giving the number of characters in the string and/or coded as follows:

I negative, horizontal page format; positive, vertical page format; zero, no title printed; equal to 100, string terminated with $.

GRAF3D - The arguments give the origin, step size and maximum value for the coordinates in units of the data. If "scale" is specified for the step size, the intervals will be scaled according to the graph size.

AXES3D - Changes the dimensions and/or draws and labels the axes

To plot a surface in z as a function of x and y:

A hidden line surface may be drawn from an external function with

SURFUN - which is called after the plot has been initialized and the workbox specified with arguments giving function name, the number of lines in each of the x and y directions and the distance between them, and the name of a previously dimensioned work space greater than or equal to the sum of the number of grid points times the distance in each direction plus 4.

or from a surface defined by a matrix with:

SURMAT (Not described here.)

To change the viewpoint:

The viewpoint can be changed by specifying the viewer position either in inches on the 3 axes or by the angle phi in the xy plane, theta in the xz plane and the distance from the origin. However the viewpoint is specified, it must lie outside the smallest sphere that could encompass either the default 1" cube enclosing the subplot or whatever workbox has been defined by AXES3D.

The default view is equivalent to:

```
call VUABS(-1.5*xdim,-1.5*ydim,+1.5*zdim)
```

When a call to VUABS is made, the arguments must be given in absolute dimensions, however.

Default for the viewing angles is equivalent to:

call VUANGL(225,0,1.5) where the angles are in degrees and the viewer distance from the origin is in inches.

The default view puts the viewer at the face of the screen with a 90 degree perspective; for display, the viewer distance should be increased to represent a more realistic perspective. In Movie 1 it has been doubled.

To define alphabets used in labels:

Two different alphabets can be defined by calls to BASALF and MIXALF with the name of the alphabet in "" as the sole argument. The alphabet specified by BASALF will be used in a text string until an open paren is encountered, after which all characters will be interpreted in the alphabet specified for MIXALF until a close paren is found, resetting the alphabet to BASALF.

```
call BASALF("standard")
call MIXALF("l/cstandard")
        .
        .
call MESSAG("this is upper case; this is (lower).$",100,x,y)
```

will result in the following message appearing at location x,y:

THIS IS UPPER CASE; THIS IS lower.

If more than two alphabets are needed in a text string, or it is desirable to specify all the sets needed for a plot at one time, up to six alphabets* may be defined by calls to MXnALF, where n is an integer from 1 to 6 and the second argument for each MXnALF is the single character, enclosed in "", that will be used to activate that character set. Obviously, care must be taken in the choice of code characters - a character used as a code will not appear as part of a message. To set up the example above using MXnALF:

```
call MX1ALF("standard",")")
call MX2ALF("l/cstandard","(")
```

General format routines:

NOBRDR suppresses a default border which is drawn around the outside of the plot; PHYSOR sets the physical origin in inches (x,y) from the lower left corner of the page; PAGE sets the physical dimensions of the page (x,y).

Integer-writing routine:

INTNO writes the integer given as the first argument at location x,y, which are given as the next two arguments, respectively. Here it is used to label each frame in sequence.

*The entire list of available alphabets can be found in the DISSPLA manuals, along with the keyboard input for the corresponding output characters.

Some Packages and Viewing Algorithms Available at LLNL

Hidden - Three dimensional package that contains the algorithms to draw three dimensional objects with hidden lines and surfaces from input data. A hidden line drawing is one which suppresses the lines which would be obscured by an opaque surface appearing in front; a hidden surface drawing is identical except that the surfaces are shaded. Solid objects are formed of polygons and suitably shaded to appear curved or faceted, as desired. Contains dithering (see below) and has a left-handed coordinate system. A complete description with instructions for use can be obtained from UCID 30057, Rev. 1.

AtomLLL - A program which draws colored, shaded and highlighted hidden-surface spheres from input data specifying the sphere colors, radii, chemical bonds (in the case of a molecule) and the coordinates of their centers. The output file from AtomLLL can be copied to a magnetic tape for input to the Dicomed D48 film recorder. Details are contained in UCRL-52645.

Display - An interactive plotting routine that combines data from several sources (including different machines) into picture elements which can be manipulated, viewed and edited interactively from a CRT, or sent out to a graphics or hard copy device. A subroutine library package called DPPAC provides Fortran callable entry points to Display. UCID-30081 contains a complete description of the program.

Dither - Converts gray-level picture data, i.e., data which creates a picture from different display intensities on a vector-type CRT, to pseudo-gray level format for output to a digital device such as a TV monitor (TMDS) or a digital printer/plotter (RJET). It may be called as a subroutine from TV80lib and Graflib, as well as from Hidden. Complete instructions for its use are contained in UCID-30144.

Grape - Postprocessor for Hidden which contains several shading routines. Uses Watkins hidden surface algorithm. Does finite element analysis on input data giving original shape elements and nodes, motions and stresses. Outputs resulting motions, stresses and displacements. Description in UCID-18507.

Before using the system you will need a user number and account number and a terminal with which to log in, as well as a box number in the printer room for receiving output. An interactive program called DOCUMENT allows one to view all or parts of most documents on line and to have the entire documents printed out. Particularly useful documents are INTROMFECC, a guide to the system, and TEDI, a description of a text editor. There is also a handbook available which contains summaries of all the utility routines.

HELP

General Consultation: (415) 422 1398

Direct questions about how to get started to (415) 422 4022

Direct questions about software to Computer Consultants, (415) 422 1544.

(The above numbers are available only during normal LLNL business hours.)

Direct questions about hardware and mainframe status to (415) 422 4283

In all the examples of computer input, file names and program names are written in upper case only for emphasis; whether upper or lower case is used depends on the output mode of the terminal.

Log on (c machine): c userno. a* groupno. ctrlzPASSWORD

Compiling, loading and running

On the CRAYs the RCFT compiler can be used to compile (from a source file called NAME), load and execute in one step as follows:

```
RCFT i=NAME,lib=graflib,go
```

The default output files taken in this example are:

binary output from compiler:	BNAME
listing file:	LNAME
map file:	MNAME
executable file:	XNAME

Or the steps can be done sequentially and/or output file names specified:

Compile:	RCFT i=NAME,b=NEWNAME
Load:	LDR i=NEWNAME,lib=graflib,x=YETANOTHERNAME
Execute:	YETANOTHERNAME
One step:	RCFT i=NAME,b=NEWNAME,lib=graflib,x=YETANOTHERNAME,go

* or b, c, d (= computer channels)

Text Editing TEDI

There are several text editors available, all of which are similar to each other and to standard computer editors. One of them, TEDI, is described briefly here; further information can be obtained by the appropriate document and from the small NMFECC handbook.

TEDI NAME

NAME is the name of the source file to be edited.

To initiate text writing mode:

aln	Add text after line n
bln	Add text before line n.

Text writing mode prompt is & To exit, type .

Line editing command:

xn Line n will appear for editing. Any character except a space will replace the character above it. Space is used for positioning - if a cursor is visible, it will be under the character to be changed; if not, the spaces must be counted to know which column will be replaced. Insertions can be made by first pressing up arrow or a caret; deletions by typing a backward slash.

Line editing prompt is a : To exit, type return.

Text manipulation commands:

tnl,n2	Type lines nl through n2.
dlnl,n2	Delete lines nl through n2.
canl,n2,n3	Copy lines nl through n2 to position after line n3.
manl,n2,n3	Move lines nl through n2 to position after line n3.
rpnl,n2;pl;p2	Replace text pl in lines nl through n2 with p2.

File manipulation commands:

cfanl,n2,n3$NAME	Copy lines nl through n2 from file NAME to position after line n3 in file being edited.
cfbnl,n2,n3$NAME	Copy lines nl through n2 from file NAME to position before line n3 in file being edited.

Formatting command:

si7,3	Sets first column to 7, increment to 3 for Fortran statements.

Command mode prompt: * To exit TEDI, type end.

Utility Routines

[] are not typed in; they indicate optional input.

FILES — Lists all user's current* files

DESTROY — Removes the files from user's space; filenames typed after a space and separated by commas.

```
DESTROY FILE1 FILE2 FILE3 etc.
```

SWITCH — Changes file name:

```
SWITCH OLDNAME NEWNAME
```

FILEM — For permanent storage of files. To use type FILEM and return or follow with space and function:

WRITE Writes files from user's space to storage:

```
WRITE FILE1
```

READ Reads stored files into user's work space:

```
READ FILE1
```

DELETE Deletes file from storage:

```
DELETE FILE1
```

REPLACE Replaces old file with new version with the same name; program will not overwrite a file by the same name unless this option is used:

```
REPLACE FILE1
```

DOCUMENT — Lists all documents available on line; gives partial listing if viewing option (v) taken; prints out entire document if printing option (p) taken. For viewing option, asks in addition to WRITEUPμ (document name) KEYWORDμ, which means the section of that writeup that you wish to view. Answering INDEX will get a listing of the names and description of each section of the writeup.

NETOUT — Transfers a file from one machine to another or to an output device :

```
NETOUT FILE1 [options] [site=destination]
```

Default site is the user terminal.

NETPLOT — A very versatile routine that can be used to plot a graphics file :

```
NETPLOT FILE1 [options] [site=destination]
```

Can also be used to plot a text file by using option text=file.

* Active files which are dropped from the system if not used for 24 hours.

Glossary

anti-aliasing	A technique consisting of varying the intensity and/or grouping pixels of a display to minimize the "staircase" pattern of a diagonal line drawn on a raster or digital device.
aspect ratio	The ratio of height to width of a frame of movie film, which equals .75.
bi-cubic patch	A complex surface calculated by a cubic polynomial in two variables which is then superimposed on a smooth surface.
bleeding	The running over of a color into another area, caused by scattered light exposing adjacent film granules.
blooming	See bleeding.
blowup	Enlargement of all or a portion of a displayed picture.
clipping	Removing the portions of a picture that fall outside the viewing area.
clipping divider	Hardware device attached to a CRT unit to do clipping.
color compensating lens	A lens used with the filters of a color film recorder that brings all the light of different wavelengths to the same focus.
computing perspective	The calculation involved in making an object in a two-dimensional display appear three-dimensional.
cross dissolve	A technique used to minimize the jerkiness seen when only a few frames represent a rather large change in position.
depth cues	A term which applies to all the techniques used to simulate depth in two-dimensional displays.
dissolve	The fading-out of a scene to end an animated sequence.
diffuse shading	Matte, rather than glossy finish.
dithering	Technique that produces grey-level data on a binary display by area averaging.
double buffering	A technique to allow a picture to be displayed and updated simultaneously without affecting the display by using two data buffers.

Glossary

double framing	Copying each frame once to smooth out the animation.
double imaging	The appearance of two objects instead of one, caused by too rapid movement.
faceted shading	Coloring each surface of a polygon a constant shade over its entire area in order to emphasize the facets.
fading	See dissolve.
finite element	A calculation which solves a problem on a complicated structure by treating a small part of the structure at a time.
flat-shading	No change in shade across the facet of a surface. Faceted shading.
footage	A series of animated frames without sound.
forward differences	A method of solving polynomials without multiplication.
fractal shading	A texturing algorithm.
frame buffers	Memory buffers that contain several frames worth of information for a CRT raster; each pixel of the display has as many bits of information as there are buffers.
gate	Aperture on an optical printer through which an image is projected onto film being exposed.
Gouraud shading	A smooth shading algorithm that uses a shade level plus a delta shade to vary the shading evenly across a surface; minimizes faceted appearance of polygonal surfaces.
gray level	A type of display in which the image is produced by variations in intensity.
gray patch	One of a series of patches of equal gray level intensity placed on one FR80 frame of a series. Used to match intensity during film processing.
hidden line, surface	An algorithm that removes the lines or surfaces in a two-dimensional picture that would be obscured if it were a three-dimensional opaque object.
keyframe animation	The technique of calculating only a few frames of a series and interpolating between them to produce the intervening frames.

liquid gate	Term which describes the process used in an optical lab to remove superficial scratches from film.
masking	A technique in which a mask is placed in front of an image to remove portions of it from the picture. Used in an optical printer to superimpose titles, logos, and other images onto original film.
matrix multiplier	A hardware attachment to a CRT that does matrix manipulation.
matte	A high contrast black and white mask.
maximum density, dmax	The maximum density of a specific piece of exposed film; varies with the development and the type of film.
Mach banding	The illusion produced by adjacent broad areas of different colors that makes them appear to be darker in the center.
motion blur	Blur caused by the displayed image moving too fast for the eye to resolve.
pixel	Picture element, the smallest unit of a picture.
pixel shading	An anti-aliasing technique.
quadratic patch	A complex surface calculated by a quadratic polynomial which is then superimposed on a smooth surface.
raster	A rectangular grid of points on a CRT screen which are turned on or off to form a display.
scan	The procedure in which a program searches the memory locations containing display data to see if a display point should be turned on or off. The scan is usually done top to bottom, right to left (as seen on the screen).
scan line	One row of display data.
self-similar figures	Pictures used to give the impression of a zoom. Each is identical in shape to the one before it but slightly larger.
silhouette smoothing	The dividing up of the edge of a polygonal surface into several smaller polygons to eliminate the angular appearence of the silhouette.

smooth animation	The technique of slowing the rate of movement at the beginning and end of a sequence to give a smooth appearance to the motion.
smooth shading	A smoothly varying shade across a surface which helps to obscure the facets of the individual polygons of which the object is composed.
surface patches	Calculations done for a small area to describe a complex surface. They are then superimposed over a simple background in order to save calculation time.
time-resolved parallax	The effect of rotation in producing an illusion of three dimensions.
vector	A line drawn by the electron gun sweeping across the phosphor of a cathode ray tube. A vector CRT creates a display by drawing such lines, as opposed to a raster device which lights up points on a grid to create a display.
warping	Distortion of a planar surface caused by round-off differences in the points defining the plane.
z buffer	A software algorithm allocating data with depth information.
zooming	The illusion of having used a zoom lens to film the action; see self-similar figures.

Appendix III Bibliography

General Computer Information

Users' Guide to the Octopus Computer Network — UCID-30048 Rev 5
Carl Schneider, Dave Thompson, Gail Whitten

Intromfecc — On-Line
Introduction to NMFECC

Summaries of Utility Routines on CDC 7600 and Crays — MFECC-020006
A. Syde, S. Fernandez, Z. Gifford, T. Girill, M. Wakakuwa

Graphics*

Becoming Comfortable with Graphics — Hewlitt-Packard Pub.
Virginia Pollack

Principles of Interactive Computer Graphics — Mc Graw-Hill
William M. Newman, Robert F. Sproull

LLL Computer Graphics Output Devices — UCID 16381
Don Vickers

Two Unusual 3-D Displays — Transactions of the American Crystallographic Society
Don Vickers

Graphics Techniques

Computer Graphics: THe Need for Graphics Design — Computer June, 1981
Ware Myers

Display of Engineering and Scientific Data — IEEE Computer Graphics and Applications vol.1, no.3, 1981
J. E. George, A. Vinberg

Computer Graphics and the Business Executive — Ibid, vol. 1, no. 1, 1981
A. Vinberg, J. E. George

Improving Management Productivity with Computer Graphics — Ibid, vol. 1, no. 4, 1981
Alan T. Paller

Designing a Good Graph — Integrated Software Systems Corp. (ISSCO) Pub.
Anders Vinberg

Choosing the Right Chart — ISSCO Publication
Alan Paller, Kathryn Szoka, Nan Nelson

* In addition to the references listed here, a further reading list may be obtained from:
The Graphics Group Bibliography — LLNL Computer Graphics Group
Janet S. Chin

Graphics Libraries

TV80lib — LCSD-436
Steven Williams

TV80: Device Independent Graphics Routines — LCSD-305
Nancy Storch

Disspla Pocket Manual — ISSCO

Disspla Users' Manual — ISSCO

Graflib Manuals

Graphics Routines (Available at LLNL)

Computer Graphics by Example — UCID 30166, Parts 1-7
Kelly O'Hair

Users' Guide to SURFACE

Dither — UCID-30144
Chris J. Benenati

MCHARTSC — UCID-30165
Janet S. Chin

Displaying Complex Three-Dimensional Objects (HIDDEN) — UCID-30057 Rev. 1
Michael J. Archuleta

Displaying the Results of Three-Dimensional Analysis Using Grape — UCID-18507
Bruce E. Brown

ATOMLLL - A Three-D Opaque Molecule System — UCID-52645
N. L. Max

Display — UCID-30081
Jeffery H. Rowe

Display: An Interactive Picture Editor — Energy and Technological Review, LLNL Jan., 1977
Jeffery H. Rowe

Movie-Making

The Role of Computer-Generated Movies in Scientific Research — IEEE Digest Spring 1979
Don Vickers

Computer-Generated Movies as a Scientific Tool — LA-UR-78-66, Los Alamos National Laboratory, 1978
R. L. Elliott

Short Films from the Topology Film Project — UCID-17900
N. L. Max

Optical Printing

Tutorial on Optical Printing UCRL-83546
John Blunden, Nelson Max

Additional References:

Beatty: REDPP	UCID-30125
Vickers: Integral Hologram as a Scientific Tool (1976)	UCID-17035
Cecil: TRIX AC A Set of General Purpose Text Editing Commands	UCID-30040
Moll:TRIX An Interactive Interpretive Language for Manipulating	UCID-30100
Moll: The TRIX Report Editor	UCID-30142
Blair: The UXD080 Graphics System	UCID-30146
Beatty: PICTURE - A Picture-Drawing Language for TRIX/RED Editor	UCID-30156

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720