Graphics software for remote terminals and their use in radiation treatment planning*

by KARL H. RYDEN and CAROL M. NEWTON

University of California at Los Angeles
Los Angeles, California

INTRODUCTION

Interactive graphics' ability to provide a meaningful interface to investigators in a variety of applied fields has been recognized for many years. Nowhere is this promise greater than in medicine and the life sciences. Levinthal's interactive program for constructing and displaying complex molecular structures is well-known. Neurath and Frey have analyzed chromosome spreads by delegating to human perception the light-pen dissection of individual chromosomes from overlapping clusters while leaving subsequent measurements and final classification to the computer. Cox and Clark's Programmed Console has introduced many radiologists to the use of computers in treatment planning. Other applications at this facility and elsewhere illustrate the great value of interactive graphics support in deriving insight from large clinical data bases, exploring complex biological models in the hope of improving treatment strategies, and developing cost-effective algorithms or special hardware for patient monitoring.

That the use of computer graphics in biomedical and other fields has not developed more rapidly has been in large part attributable to two factors, the high costs associated with hardware and development of applications programs, and the general unavailability of graphics terminals that can operate remotely using common carrier communications facilities. The latter capability is not needed when modest computational requirements enable support by a dedicated small computer, as in the case of the widely used Programmed Console. But, as illustrated by other of the projects mentioned above, many biomedical applications require access to a level of computer support which is most economically provided on a shared basis. The recent commercial availability for less than $20,000 of a terminal (IMLAC) which has many characteristics of the higher quality graphics systems and which uses common carrier telephone facilities to communicate with its host processor, has motivated a substantial investment in systems support at UCLA's Health Sciences Computing Facility (HSCF).

The setting for this work will be described, previous efforts in graphics and the development of an operating systems environment appropriate for supporting a multi-terminal remote graphics system. Subsequent discussion of a set of interactive graphics programs that are being developed for radiation treatment planning will illustrate some of the problems that must be faced when a graphics terminal is to be supported by low-bandwidth communications. After describing the related hardware, current and projected software developments will be discussed with reference to the foregoing applications.

PREVIOUS DEVELOPMENT OF GRAPHICS AND OPERATING SYSTEMS AT HSCF

HSCF is maintained by the National Institutes of Health to explore and develop computer-implemented analytical support to research in biology and medicine. The BMD statistical package programs, TORTOS operating system, and other developments at this facility have originated in response to explicit needs of biomedical researchers.

TORTOS (Terminal Oriented Real Time Operating System) was developed in response to the variety of needs that prompt biomedical researchers to share access to a large computer. It has been designed to take

* This work is being undertaken at the Health Sciences Computing Facility, UCLA, sponsored by NIH Special Research Resources Grant RR-3.
advantage of the large high-speed core (2000K bytes) and computational speed of the 360/91 processor, making these resources available to the facility’s users with as few constraints as possible on the types of programs accommodated. The terminal user has at his disposal the full facilities of a standard OS/MVT system as well as the specialized interactive functions developed here to aid conversational access to the system (monitor programs, a convenient file service, input/output interfaces, text editor, a FORTRAN compiler, and a library of statistical and biomedical applications programs). The system concurrently supports local and remote batch operations. Its versatility is exemplified by the wide range of uses to which it is being put. These include multivariate statistics, computer-aided instruction, management information systems in the Biomedical Library, biological modeling, and several clinical applications. Although not presently in use, the system has supported a high speed, real time link to an SDS 9300 located in UCLA’s Brain Research Institute. Since the remote graphics terminals are time-shared as an extension of the conversational terminal system, they have access to all facilities developed for the latter.

For the past five years, we have developed and used-tested systems to make the IBM 2250 graphics terminal accessible to biomedical programmers. Interactive graphics interfaces are intended to be meaningful to the scientists who use them. A growing number of biologists and physicians are capable FORTRAN programmers. Many of this facility’s medically or biologically oriented graphics programs have been developed by such people, using our FORTRAN callable subroutines, GRAF (Graphic Additions to FORTRAN) or a similar interface, PL/OT, developed for use with PL/I. These include programs for data analysis and statistics, teaching, image processing, modeling of cellular and physiological systems, interactive retrieval, simulation of disease propagation in adjacent urban environments, design of coronary-care and obstetrical monitoring systems, and radiation treatment planning.

The GRAF programmer has at his disposal routines for generating the primitive elements of an image (blanked and unblanked points, lines, and alphanumeric characters) which are combined into a structured display file whose basic element is called a display variable (DV). Additional routines are provided for combining the DV’s into display frames for presentation on the terminal’s screen. There are also facilities for servicing the input devices associated with the graphics terminal (alphanumeric and function keyboards and light-pen) and for character-string manipulation.

REMOTE GRAPHICS SUPPORT TO RADIATION TREATMENT PLANNING

Distributed-processing capabilities being developed for the GRAF subroutines designed for terminals operating over low-bandwidth communications systems are discussed in later sections. One of the biomedical projects whose needs influence this work, a set of programs for radiation treatment planning, is described here. Preliminary program development on the IBM 2250 has been reported. 

Previous work in radiation treatment planning

Radiation treatment planning is among the earliest established areas of computer support to medical practice. Stovall’s recent bibliography provides access to most of the formal publications in this field and is recommended as a basic reference for this section.

The radiotherapist’s primary concerns in treatment planning may be briefly summarized as follows: He has at his disposal radioactive sources that can be placed within the body and beams of radiation that can be directed to it from various angles outside. He seeks a combination of either, or sometimes both, so that the resulting spatial distribution of dose delivered throughout the tumor volume is large compared to that delivered to normal tissues he wishes to spare. He also desires a relatively homogeneous distribution throughout the tumor volume itself. Fear of tumor regrowth in undertreated areas, “cold spots,” dominates the latter concern, but excessive tissue breakdown in hot spots also is to be avoided. Either formulas inherited from research on the interaction of radiation and high-energy particles with matter, or tabulated empirical measurements of fields surrounding sources or beams in water or other materials of known composition, enable him to compute the desired dose distributions. Palpation, scanning or other techniques are used to estimate the location and extent of the tumor. Special radiographic techniques such as tomography help to locate the boundaries of internal organs to be spared or whose tissue densities are to be taken into account when the dose distributions are calculated.

When done by hand, even the simplest calculations, such as the dose distribution for two or three superimposed beams in a homogeneous unit-density medium, are extremely tedious and tend to be undertaken infrequently and reluctantly where computer support is not available. Thus in many places radiation therapy adheres to longstanding conventional prescripations that have proven relatively effective for the “average
patient” given the constraint that the risk of notable side effects in all is to be vanishingly small. Computer-assisted treatment planning seeks to individualize therapy, to help the therapist take every possible advantage of a particular patient's tumor location and body geometry. It also encourages him to explore new general treatment strategies which might, even in places that do not use computers, become part of the conventional armamentarium of treatment plans.

Two types of computer support for treatment planning are regularly used in a small but noteworthy number of treatment centers. Some excellent batch programs for external-beam therapy have been tested by many years of use. Representative of these are Theodor Sterling’s programs (University of Cincinnati and Washington University, St. Louis), Jack Cunningham’s (University of Toronto), and those developed under the direction of John Laughlin (Memorial Hospital, New York). The latter are widely available via remote typewriter terminals. The earliest program extensively used for implant therapy (placement of sources within the body) was developed by Robert Shalek and Marilyn Stovall (M. D. Anderson Cancer Research Hospital, Houston). Both the three-dimensional complexities of implant therapy and the corrections for tissue inhomogeneities found in the batch programs for external beams require major processors.

The Programmed Console developed by Jerome Cox, Wesley Clark, and their associates (Washington University, St. Louis) provides clinicians an economical interactive graphics system for treatment planning. It permits simple beam superimposition in a homogeneous body whose outer contour is specified to the computer by a rho-theta digitizing scribe. Program modifications are relatively difficult to introduce because of full utilization of the small processor dedicated to the system, and a number of the capabilities available in batch programs understandably are not present. However, the Programmed Console has demonstrated by wide acceptance of the radiotherapist’s strong preference for a hands-on graphics system which provides him rapid graphical dose distribution feedbacks for a sequence of conveniently repositioned superimposed beams. Other small-processor graphics systems have been developed in the United Kingdom and elsewhere.

Remote graphics radiation treatment planning**

Clearly, one wishes to combine as economically as possible the advantages of interactive graphics and major processor support, developing programs having the greatest possible degree of exportability. IBM 2250 programs utilizing GRAF have run on a variety of IBM 360 computers operating under OS, and the new version of GRAF designed for the IMLAC terminal has been organized and will be documented in a manner that should facilitate its rewriting for other processors. Immediate exportability is enabled by the IMLAC terminal’s low price and its ability to operate over conventional telephone lines.

The first two members of our package of graphics programs to support radiation treatment planning have been selected as follows: First, one wishes to take fullest possible advantage of excellent batch programs that have been tested by years of use. The initial program in this category is a graphics adaptation of Memorial Hospital’s external-beam program. RADCOMP III, the most recent version of M. D. Anderson’s program, and Sterling’s program will be next. Although there is some overlap among these, it is desirable to offer the different programs that have won various radiotherapists’ confidence. Second, we want to explore the special advantages of our system. For this we have chosen a three-dimensional implant application, the widely used Fletcher-Suit intraepptic method for treating cancer of the uterus.

Several capabilities are common to all programs: input of a brief information record for each patient, routines for providing a proper scale and assessing its

** We wish to acknowledge the very valuable collaboration of Richard Nelson, City of Hope, Duarte, California.
uniformity throughout the CRT display whenever graphical information is to be specified by light-pen, and contour plotting routines for displaying dose distributions.

1. The external-beam program. When users wish to take advantage of the Memorial Hospital program’s capability to correct for tissue inhomogeneities, they must specify contours and densities for the various organs. They may do so by taping a transparent drawing over the CRT face, using the light-pen for a point-
to-point specification of each contour. (A continuous light-pen tracking capability is available on the IMLAC and on some other IBM 2250 models.) It is easy to revise or restart contours. The expense of additional input scribing equipment is avoided. This is much simpler to execute and easier to verify visually than providing the numerical contour specifications required for the batch program. Further burdens of coding input information are relieved as the user next responds to a conversational program that requests information concerning the beams to be used and how they are to be directed (i.e., from a given angle, rotating about a given point, or oscillating within specified angular limits). Brief notations for each beam are added to the body-contour diagram (see Figure 1). Dose-distribution in the plane containing the beams then is displayed. This may be filed for later comparison with other treatment plans.

2. The intracavitary program for uterine cancer. In afterloading approaches, hollow instruments (the afterloaders) are placed securely at various locations in the tumor area. Subsequent introduction or removal of radioactive sources within the lumens of the afterloaders may therefore be accomplished without further discomfort to the patient or disturbance of the tumor-source geometry. The Fletcher-Suit afterloading system for uterine cancer comprises a long curved tube which is introduced into the uterine cavity (the tandem) flanked laterally by two source holders (the colpostats) externally adjacent to the uterus. Each of the latter can hold a single radioactive source, and a sequence of sources can be spaced along the central axis of the tandem. Frontal (A-P) and lateral radiographs are used to locate the afterloaders.

A conversational program facilitates light-pen specifications of afterloader locations, taking full advantage of their known geometries. The user may elect tradeoffs between rapidity of input and the security of checks enabled by obtaining redundant geometrical information (see Figure 2). He also may control economy vs. quality of computation by such means as specifying the length of segments to be used to approximate linear radiation sources of uniform density. For instance, an economical initial exploration which approximates each linear source by three segments can be followed by a computation of higher quality which assumes a segmentation of nine or ten. Dose-distribution displays may be requested for any A-P, lateral, or transverse plane (see Figure 3). Inventories are maintained for the afterloaders and sources.

Arrangements are being made for live demonstration of these programs on the IMLAC terminal, in addition to motion-picture illustrations of their use.

Figure 3—Isodose plot for uterine treatment: Dose distributions may be requested in any A-P, lateral, or transverse plane. This display may be transmitted line-by-line as it is being computed. An occasional segment may be missed by the economical algorithms used here.

Possible implications of graphics in the acceptance of treatment-optimization programs

Linear programming and other approaches have been suggested for optimizing spatial dose-distributions, and one is being used regularly in a small number of radiotherapy centers in Great Britain. Graphical demonstrations tend to be more convincing than mathematical proofs to the average clinician, and they help him to bring his experience to bear on criticizing the outcomes of using various criteria for optimality. One therefore anticipates that the present conjunction of an interactive graphics terminal with a processor capable of performing optimization calculations will encourage serious investigation, improvement, and perhaps eventual acceptance of more optimization procedures in radiotherapy.

Implementation over low-bandwidth communications systems

Transmission over conventional telephone lines poses serious response-time problems for the more complex graphical displays of organ contours and dose-distributions. Subsequent recall for display purposes rather than initial point-to-point specification is of main concern in the organ-contour display. However, methods currently
being explored to expedite transmission of a retrieved display may simplify the initial input as well. Advantages of the interactive approaches that are possible for anatomical specifications or the input of graphical field-distribution data for sources and beams cannot be realized for the computed dose-distributions, since the latter are initiated at the host processor. Research on these problems in graphical representation is being pursued initially on the IBM 2250 while the remote terminal version of GRAF is being developed and tested on the IMLAC. However, both efforts assume a distribution of processing between the terminal and host computer which has not been feasible on the IBM 2250. This work is described in a later section, and additional presentations will be made at the time of the conference.

REMOTE GRAPHICS HARDWARE

The remote graphics system consists of three major hardware components: the host processor, a 360/91 computer with 2 million bytes of high-speed core and 672 million bytes of secondary direct-access storage; the graphics terminal, a mini-computer (PDS-1) manufactured by the IMLAC Corporation; and the data link between the two processors. While the typical graphics program does not normally require the resources of a computer like the 360/91, it does permit the computer to simultaneously service far more graphics terminals than could be handled by a less powerful machine. In addition, the occasional heavy demands for arithmetic processing can be absorbed without severely impacting other concurrent activities,
for as at most computer centers, graphics is only one aspect of the computer's total load at HSCF.

The development of a remote graphics terminal is not a new idea, nor is using a small computer to overcome the restrictions of low-bandwidth data links, as has been done elsewhere. There are also available today at least five different storage-tube terminals which have a serial-communications interface that permits them to operate from a remote location without a mini-computer. Storage tubes do not need local refresh memories and their display capacity is limited only by the resolution of the screen. By contrast, the display capacity of refreshable tubes is limited by the size of the local refresh memory and the drawing speed (refresh cycle) of the deflection system. However, the storage-tube terminal's limited capacity to selectively erase one component of a display from the screen requires compromises between response time and the ideal presentation of information to be displayed. In order to modify one component, the screen must be erased and the entire modified display retransmitted to the terminal. In one storage-tube system it is possible to erase information from the screen by retransmitting the orders which created the display, but this also erases other components of the display frame which may have intersected the portion which was erased. Lack of a selective erase can be overcome to some extent by the use of a local computer which can regenerate the image; this, however, nullifies much of the cost advantage which the storage-tube terminal otherwise enjoys. The GRAPHIC-II system developed by Bell Telephone Laboratories is similar in many ways to ours. There are, however, some major differences in approaches to optimizing the use of the data link and to distributing tasks between the remote and host processors. The PDS-1 is one fourth the cost of the PDP-9 system used in GRAPHIC-II. In part, this reduction in cost is a result of technological advancement; however, much of it is due to the fact that the PDS-1 was designed specifically as a low-cost display computer.

The IMLAC PDS-1 (see Figure 4) computer is a dual processor machine with each processor sharing the same memory. The main processor executes an instruction set which is typical of most small computers (ADD, SUBTRACT, BRANCHING, LOGICAL and INPUT/OUTPUT operations) while the display processor, which is controlled by the main processor, executes a set of display instructions that generate the image on the CRT (see Figure 5). The display image is drawn using incremental vectors; each incremental order (8 bits) can move the CRT beam in one of 32 directions for a distance of 0-9 raster units (a raster unit is 0.011 inches). Line segments, characters and other graphics figures are generated by display subroutines composed of incremental orders. An eight-level hardware push-down is provided for nesting display subroutines.

The principal disadvantage of the incremental drawing technique used by the IMLAC is the slightly jagged appearance of lines or curves that results from the 32-direction restriction on elements composing them. This can be a serious problem if high-quality images are required. For instance, depth cues of the perspective transformation might be blunted when three-dimensional images are being generated. However, in the majority of medical applications, this is not a serious problem and the economies that result from the use of incremental drawing seem to outweigh the disadvantages.

The choice of a data link between the graphics terminal and the host processor was dictated by universal availability, low cost of modems and interfaces, and speed, in that order of importance. We chose to use a serial-synchronous communications format employing switch network telephone facilities at 2000 bps (WE-201A type modems). However, while meeting our requirements of availability and low cost, the transmission speed represents what can only be considered a minimal level. A major component of the programming support is devoted to overcoming deficiencies of the data link.

Figure 5—IMLAC graphics display terminal: Light pen and optional programmers console are shown to the left of the keyboard and display
REMOTE GRAPHICS SOFTWARE

Programming systems support for the remote graphics terminals consists of a control program for the terminal computer, the host processor control program, and the applications programming “language.” The latter two components are 360/91 programs while the first executes in the PDS-1 computer. Application programs are written in FORTRAN using a collection of subroutines called GRAF which were originally designed for use with the IBM 2250 and whose external format has been preserved in the remote graphics implementation. Thus, use can be made of the large number of programs written using GRAF at UCLA and elsewhere over the past four years. Although an emulation of the 2250 would have provided access to more existing programs, it would have compromised effective use of the data communications facilities.

Strategies for overcoming the limitations of low-bandwidth communications systems

Transmission over conventional telephone lines poses serious response problems for the more complex displays (see Figures 1 and 3). In Figure 1, the organ contour display, the main concern is with the subsequent recall of the image rather than its initial point-to-point specification whereas in displays like Figure 2, response is poor because the terminal fails to react quickly to simple interactions. The logical and arithmetic capabilities of the remote processor offer several means of improving response. Data compression can increase the information density by removing extraneous bits from the graphic data structure or bits whose meaning is implicit from the context in which they are used. Another method is to organize the graphics data so that inactive static segments of the display are saved and recalled into the active display by the transmission of a single command. Labels, grid lines, indexes, and explanatory text are examples of data which are fixed throughout the course of the typical graphics program and need only be transmitted once when sufficient remote processor storage is available. Response time is determined by the variable display data.

Approaches that decrease the number of bits transmitted to specify graphical or other data will be discussed in the next section. The portions of GRAF initially developed for remote IMLAC terminals have emphasized storage-management and look-ahead capabilities that facilitate terminal response by avoiding retransmission of graphical information that may be displayed repeatedly and by anticipating transmission of that information which is most likely to be used next.

Unless the remote computer is augmented by a peripheral storage device, the static display data which can be retained generally will not exceed two or three display frames. The added cost of direct-access storage caused us to prefer extending the storage capacity of the standard terminal by formulating a procedure which retains those inactive segments of the terminal’s display buffer which are most likely to be reused when it becomes necessary to overlay some segment in order to accommodate a new segment. This procedure is primarily a storage management technique and has been augmented by look-ahead facilities which are designed to anticipate the requirements for new display segments and transfer them during intervals when the data link would otherwise be idle.

New GRAF storage-management facilities

Implementation of the storage management and look-ahead facilities is largely unknown to the GRAF programmer although he has some control over the transmission of graphics data to the remote processor.

As in the IBM 2250 version of GRAF, the basic element of the graphics data structure is the display variable (DV), which is composed of the primitive elements that generate the display. The DV is formulated in the host processor by graphics subroutine calls, for subsequent transfer to the display buffer, that portion of the terminal’s memory which services plotting.

The display buffer of the remote processor is divided into primary and secondary display lists. The secondary display list entries correspond to a subset of the DV’s in the host processor. The primary list consists of pointers (display subroutine jump instructions) to each entry in the secondary list which is currently being displayed on the CRT screen. DV’s are added or removed from the screen by adding or removing appropriate pointers in the primary list. A display frame is composed of one or more DV’s which may contain their own screen coordinates, or the position which the DV occupies may be determined by positioning orders in a DV which “calls” it. The DV can be considered a display subroutine, and in the instance of a relative DV (one which does not contain absolute positioning orders), only one copy of the DV is stored within the remote graphics processor.

The best response characteristics will be obtained when each frame is segmented into DV’s such that the static information is separated from the variable data, and when display components which are used re-
peatedly are each declared to be unique DV's rather than combined into a few DV's. Thus, the user can improve response characteristics not only by partitioning each display frame into fixed-format and variable DV's but also by further subdividing the fixed-format portions to take advantage of relative DV's that are to be used repeatedly. For example, in a single session with the computer, the radiotherapist probably will explore up to three or four treatment approaches for each of several patients. There will be delays when DV's guiding conversational input are initially called, but use of the program thereafter would, for the most part, flow as freely as on the IBM 2250. The saving is especially notable for the grids supplied to effect input of graphical data. One coarse grid covers the entire screen, but for more precise description of curves a finer grid patch is moved to desired locations on the large grid. Treating the patch as a relative DV, only the transmission of two coordinates is required to reposition it. Because of frequent use, DV's representing both types of grid would have a high probability of remaining on the secondary list of the terminal's display buffer.

All of the decision making and tables for managing the remote processor's display buffer are provided by the host processor. The graphics processor's storage management functions are limited to unpacking the transmission records, relocating addresses, and moving entries within the primary and secondary lists. When a new DV is to be transmitted to the remote terminal, its storage requirements are determined by the host processor and it is assigned a relative location within the display buffer. If sufficient free space is available either in one contiguous block or in multiple free blocks, the DV is placed in this region. When a single region is not available, commands to compact the display buffer are transferred ahead of the graphics data. Normally, all free space will be in one contiguous block, since the host processor's storage management routines use idle time on the data link to transfer the commands necessary to compact the secondary display list. Upon receiving the DV, the remote processor unpacks the data and relocates all relative addressing information into absolute display buffer addresses. When members of secondary display lists are moved from one location to another to compact the list, all absolute addressing information in both lists is again relocated.

When insufficient free space exists for a new DV that is to be transferred to the terminal, the required space must be created by some combination of using free space and removing secondary display list entries. The criteria used for choosing which blocks(s) in the secondary list should be deleted attempt to reduce the chance of deleting DV's which could be used in a subsequent display while also minimizing response time, since further action by the terminal user is suspended until a successful transfer enables the DV to become part of the active display. Three decision levels are used: (1) The system first tries to find a contiguous set of free or inactive blocks within the secondary display list. The inactive DV's which are chosen are those with the lowest probability that they will be reused. The assignment of these probabilities is described later. Contiguous sets are sought in order to avoid an additional pause to transmit commands required to compact the secondary display list. (2) When a contiguous set has not been found, the free blocks and the inactive DV's with lowest reuse probabilities are collected into a list, weights assigned, and a good set chosen to be deleted. The secondary display list subsequently must be compacted. The weight assigned to each DV is a function of its size, its position in the secondary display list (which determines the delay necessary to compact the list if this DV is deleted), and the reuse probability. The optimum set, that which provides enough space while minimizing the summed weights of DV's to be removed, can only be chosen while examining all possible combinations. Since this would burden the host processor, an alternative, approximate stepwise procedure is used. First, the list is sorted by weights and partitioned into two groups. Initially, the first group will contain the first \( n \) elements of the list whose total size is sufficient. The procedure then compares the sum of the weights of each pair of elements in the first group with the weight of an element of the second group. An element in the second group replaces the pair in the first when its weight is less and its size is greater than or equal to the corresponding pair in the first group. Similarly, the process is repeated by comparing combinations of three elements in the first group with pairs and single elements of the second group. We have found that continuing the search beyond three combinations does not yield a significant improvement. (3) If a check made prior to pursuing (1) and (2) reveals that the total free space and that occupied by inactive DV's is less than the space required, the operator is permitted to delete DV's from the active display by using the light-pen. This final means of obtaining the required space in the display buffer is provided primarily for program debugging, permitting the program to continue when it otherwise would have been aborted.

Look-ahead facilities

The communications link between the graphics terminal and the host processor is normally idle for several
seconds between each interaction. During this interval, the user is determining his next step. This suggests that if the host processor has a reasonable probability of predicting the user’s choice, the effective transmission speed of the data link could be increased by using the idle time to transmit the graphics data most likely required for the next display frame. The display program can be represented as a discrete Markov chain whose states are all the possible display frames which the program may generate. In GRAF, the states are defined by the DV’s which generate the display frame. The transition from one state to the next occurs at each interaction (light-pen detection, keyboard entry, or terminal processor interrupt).

To predict which state is most likely to occur next, we form a transition probability matrix

\[ P(i^*, n+1) = P_i^* \{ X_{n+1} = j \mid X_n = i \} \]

\( P(i^*, n+1) \) is the probability that \( j \) will be the display program’s next state given that \( i \) is the current state.

Each time the program enters a waiting state for some form of remote processor interrupt, the transition probabilities of the current state are examined and the DV’s corresponding to the most probably next state of the program are transferred to the remote processor to the extent permitted by time and space in the display buffer. Not all of the DV’s corresponding to the next state can be transferred in advance since some may not be defined until after the operator interaction is completed and others, although fully specified, may not have been generated in advance.

The state matrices are of two types: When the program is being executed for the first time, no information about the transition probabilities exists. All probabilities are assumed equal unless the programmer specifies otherwise. The second type of state matrix represents a typical use profile for the program which has been accumulated through many executions of the program. Even when an extensive use profiles exists, considerable improvements in the predictions can be made by dynamically updating the transition matrix so that it will more accurately predict the current use of the program. Provisions are made for each user to save and update his own transition matrix to tailor the program’s response to his own needs. For example, new users of the intracavitary radiotherapy program will tend to seek more explanatory material and to elect more conservative approaches for locating the afterloaders. They will tend initially to use the transition matrix provided by the system, changing to their own when familiarity with the program motivates expedition of shortcuts and other preferred routings through the program. Thus, the system’s matrix for each program will tend to remain oriented toward its most likely client, the new user.

**Data-compression techniques**

Processing capabilities of the graphics terminal enable unpacking of information transmitted from the host processor in a variety of condensed formats. While computations required for condensation can be complex without taxing a fast host machine, unpacking must be efficient enough to provide the desired advantage over unpacked transmission formats. The radiotherapy programs have directed attention to the problem of compacting information required for specifying graphical displays. Investigations proceeding on the 360/91 and 2250 while basic GRAF software is being tested are mindful of the two types of local graphics interfaces provided by the remote terminal: (1) specialized function generators such as those supporting curvilinear display generators, the light-pen tracking routine, routines which can translate, scale, and window display variables; and (2) routines which are written in the IMLAC PDS-1 machine code by applications programmers. The routines for both types of interface are assembled by the host processor and stored in a library to be dynamically loaded into the remote processor as they are required by the graphics programs. For purposes of memory management, the relocatable PDS-1 programs are considered to be display variables. Remote processor programs can extend the capability of the entire remote graphics terminal system, particularly where display dynamics are involved.

More general, conventional approaches to parameterized representation of graphical data are being explored initially, e.g., conic sections \(^9\) and families of exponential curves, \(^9\) as alternatives to piecewise representation by straight-line segments. The combined processing capabilities of this system permit additional approaches to be explored without the immediate constraint that they be implementable in hardware. For instance, for specification of organ contours or the distribution of fields around radioactive sources or X-ray beams, the approximation of large segments by French curves visually fitted at the terminal is being investigated.

Graphical information originating at the host processor, such as the dose distributions calculated for a given patient, cannot benefit initially from interactive approaches such as the latter. However, since considerable computation is required for each point in the dose-distribution field, procedures that transmit graphical
Figure 6—Polar transform with different expansion moduli: Among approaches being explored to compact patient anatomical information for subsequent rapid transmission is representation of an individual's configuration by transforms mapping it into standard anatomical cross-sections. The outer contour on the left (non-anatomical) figure is being mapped into that shown on the right by expansions around a central point that assume different "expandability" of various domains. In this case, the outer (fat) layer is assumed to vary more than other "tissues."

information as the doses are being calculated alleviate much of the transmission delay. They have the additional advantage of enabling the radiotherapist to economize by terminating computations when the information displayed suffices for decision making. The contours shown in Figure 3 were generated line by line. The line at which computations are to commence may be indicated by the user. Another routine is available which computes and displays contour by contour.

An interest in transforms of related sets of graphical data is aroused by the awareness that a number of radiotherapists estimate internal-organ location by adjusting standard anatomical atlas cross-sections to measurements of the patient's external contour. We are exploring an improved version of this approach which allows different size-preserving factors for various tissues to be observed while transforms are being applied (see Figure 6). The objective is to reconstruct an adequate display of an individual's anatomical cross-section, using processing capabilities of the graphics terminal, locally stored standard cross-sections, and a small number of parameters characterizing a previously determined resultant transform. The latter would be obtained initially at the graphics terminal, as the radiotherapist seeks by a number of transforms on the entire cross-section or on individual organs to achieve an adequate correspondence between a transformed atlas cross-section and that determined for the patient by tomography or other methods.

By storage-management and look-ahead facilities, we have sought to provide users the most economical terminal possible. However, should radiotherapists wish to add a disk at the terminal for more complete anatomical atlas storage or capacity for additional stand-alone applications on the IMLAC, it would be worthwhile to investigate transferring to the terminal processor a stand-alone capability for many conversational portions of anatomical and treatment specification, to reduce subsequent communications charges. These program developments would be expedited by a 360/91 FORTRAN compiler that generates object code for the IMLAC.

Performance measurements

At this writing, we have not had enough experience with user programs to give any quantitative assessment of the remote graphics terminal's performance as compared to the 2250 terminal. There are areas where further development is already indicated. Decreasing response time by facilitating programmer's access to the remote computer through a compiler or by the addition of hardware components are areas to which we are now turning our attention.
In order to evaluate performance of the system and to quantitatively measure the effect of changes in the hardware and software, an extensive data collecting facility has been built into the programming packages. The programs continually log information regarding response time, data link utilization, storage utilization, display variable characteristics and other parameters which are believed to relate to system performance. This information is printed for the programmer at the end of each graphics job and is accumulated in a historical file for later analysis. Being provided with performance information regarding the execution of his graphics program, the user is likely to take an interest in those aspects of the program which affect its performance and to take the necessary steps to improve his program. The data gathering routines of the graphics system can also be instructed by an execution time parameter to collect a complete profile of a graphics job to the extent that its execution can be reproduced by using the profile as an input source instead of the terminal. This is particularly useful in debugging the system by providing a reproducible set of test jobs which can be run over and over until an error is found and corrected or the desired performance improvements have been made.

REFERENCES

1 C LEVINThAL
Molecular model building by computer
Scientific American Vol 214 No 6 pp 42-52 1969
2 P W NEURATH B L BABLOUZIAN
T H WARMS et al
Human chromosome analysis by computer—An optical pattern recognition problem
3 H S FREY
An interactive program for chromosome analysis
Computers and Biomedical Research Vol 2 pp 274-290 1969
4 W F HOLMES
External beam treatment-planning with the programmed console
Radiology Vol 94 No 2 pp 391-400 1970
5 W J DIXON
Annual report 1968-1969
Health Sciences Computing Facility University of California Los Angeles Appendix B and pp 83-85
6 T L LINCOLN
The clinical significance of simulation and modeling in leukemia chemotherapy
7 P M BRITT W J DIXON R I JENNrich
Time-sharing and interactive statistics
8 User's guide for GRAF: Graphics additions to FORTRAN
Health Sciences Computing Facility University of California Los Angeles 1968
9 A HURWITZ J P CITRON J B YEATON
GRAF: Graphic additions to FORTRAN
10 R C UZGALIS H FREY
PL/OT, Graphics Subroutines for PL/1
Health Sciences Computing Facility University of California Los Angeles 1969
11 C M NEWTON
Remote graphics radiotherapy treatment planning
12 M STOVALl
Bibliography of computers in radiation therapy
13 G K BAHR J G KEREAKEIS H HORWITZ R FINNEY J G ALVIN KOODEE
The method of linear programming applied to radiation treatment planning
Radiology Vol 91 pp 656-667 1968
14 C S HOPE J S ORR
Computer optimization of 4-Mev treatment planning
15 C M NEWTON
What next in radiation treatment optimization?
16 C CHRISTENSEN E N PINSON
Multi-function graphics for a large computer system
Proceedings of the Fall Joint Computer Conference Vol 31 pp 697-711 1967
17 I W COTTON F S GREATOReX
Data structures and techniques for remote computer graphics
Proceedings of the Fall Joint Computer Conference Vol 33 pp 533-544 1968
18 M D RAPKIN O M ABU-GHEIDA
Stand-alone/remote graphic system
Proceedings of the Fall Joint Computer Conference Vol 33 pp 731-746 1968
19 L G ROBERTS
Conic display generator using multiplying digital-analog converters
20 M L DERTOUZOS
Phaseplot: an online graphical display technique