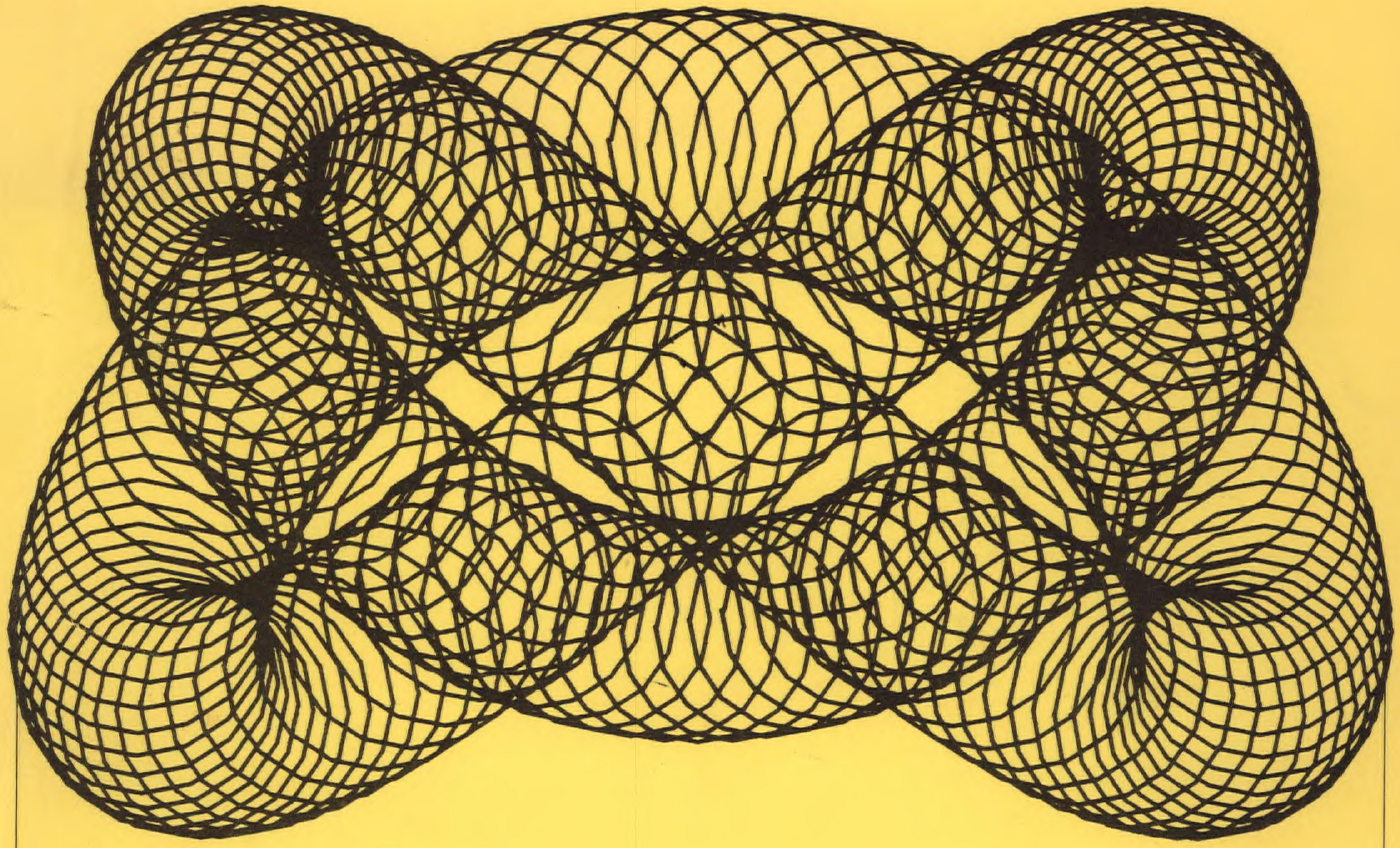


REMOTE GRAPHICS

The Communication of Structured Images

REMOTE GRAPHICS

The Communication of Structured Images

by

JAMES K. CAVERS

Dept. of Systems Engineering and Computing Science
Carleton University
Ottawa, Canada

a report prepared for
Directorate of Data Systems and Networks Research and Development
Technology and Systems Research and Development Branch
Department of Communications, Government of Canada

Contract OSU76-00109 Scientific Authority: M Sablatash

February 8, 1977

ABSTRACT

The report describes a sophisticated type of image communication system which is similar to computer graphics, and is distinct from the more familiar facsimile transmission. These remote graphics systems are attractive in several economic and social areas, ranging from business management to entertainment. Although their use is not yet widespread, it should increase sharply in the near future. Some of the implications, particularly in entertainment, are disturbing.

Remote graphics, a relatively new area, is concerned with the organization of a computer graphics system in which major components are separated by serial communication lines. A main theme of the report, therefore, is the tradeoff between terminal complexity and communication load. Technological advances in semiconductor devices and in communication systems (in particular commercial packet switching) are assessed with regard to their suitability for remote graphics work. It is shown that the terminals are becoming more powerful and less expensive. The discussion on data compression for reduced communication load points out the need for certain hardware functions not yet implemented by graphics manufacturers. The organization of the software is less advanced and several outstanding questions remain.

REMOTE GRAPHICS

The Communication of Structured Images

CONTENTS

CONTENTS (continued)

LIST OF ILLUSTRATIONS

1. INTRODUCTION

This is a report on present and future developments in a particular type of image communication -- that of structured images. It is distinct from the more familiar facsimile transmission in its objectives, capabilities and techniques. An alternative, but less accurate, characterization of the material presented is as remote computer graphics, in which the various functional components of a graphics system are partitioned into two or more subsets which are connected only by relatively low speed serial communication lines.

The main theme of the report is the effect on structured image communication of two significant processes of information technology: the advances in semiconductor manufacturing techniques and the introduction of commercial packet switching. These forces will bring graphics out of the lab into the board room, the classroom and the living room.

1.1 STRUCTURED IMAGES

What constitutes an image? At the lowest level of description it is simply the pattern of arrival rates and energies of photons striking the individual rods and cones of the viewer's retina. From this viewpoint it would appear that an image is fundamentally discrete in time and space, as well as structureless. Out of this photochemical turmoil, however, the viewer perceives a continuum of intensity/colour, space and motion and manages to isolate components of the image as separate entities. He imposes order on the scene. The extent of the roles of heredity and experience in the development of this sophisticated process is of continuing interest to psychologists, neurophysiologists and others. It is generally conceded, however [1.1-1.3], that some functions, such as edge detection, motion detection and categorical colour perception are "wired in", either immediately behind the retina or in the lower levels of the cortex. These functions are "primitives" of the viewing process. At a still higher level, clusters of primitives, and perhaps raw input, are "recognized" as separate implicitly-named visual entities (e.g. a square, a mummy, or a doggie). The importance of viewing, as perception at a distance, is witnessed by the size of the visual cortex , and the many phrases which equate seeing with cognition (e.g. the author's use of "viewpoint", "it would appear", and "is witnessed by" above).

There is a strong analogy between the multiple levels of human image perception and the more rigid descriptions used in machine generation and transmission of images. Consider the problem of describing Figure 1.1 (from [1.4]) and Figure 1.2. At the lowest level, one could list the intensity/colour values at a set of preselected points (called picture elements of pels), usually located on an equispaced rectangular grid. This will be called the "unstructured" description, and is the method used in conventional facsimile transmission.

As with visual perception, the first step in structuring a description is the definition of certain primitives; typically, these are segments of straight or curved lines ("vectors" or "arcs"), points, filled-area disks and rectangles,or alphanumerics. The set of primitives used in a particular application is determined largely by the reconstruction hardware available. A structured description at a low level consists of a list of primitives with qualifying information such as intensity/colour, size, position and orientation ("attributes"). Figure 1.2, for example, could be described as a collection of straight line segments, each of which is qualified by the 4-tuple of end point coordinates. It is worth noting, incidentally, that the structured description subsumes the unstructured description as a set of lines, points or rectangles, each the size of the display screen granularity.

At a higher level of structuring, complexes of primitives can be identified by name; these are usually termed "graphic entities" or "subpictures". One way of describing Figure 1.2 is in terms of subpictures such as "window", "roof", "door", etc., each further defined in terms of primitive line segments. Entities can also consist of a cluster of other entities which are qualified by their attributes of intensity/colour, size, orientation and position. "Small house" and "large house" are examples. The precedence relationships implicit in an entire scene form a tree or lattice structure, as illustrated in Figure 1.3.

How many bits are required to describe Figure 1.2? As the number of bits increases there is a corresponding increase in communication cost and delay, so the question is certainly relevant. The answer, of course, is that the question is meaningless, since it depends on the amount of prior knowledge available at the receiver. The structured description is a way of organizing such prior information.

Consider first the lowest level of structure, the list of qualified primitives (which, being common to both transmitter and receiver, are a form of prior information). In the case of arc or alphanumeric primitives, the gains relative to the unstructured description are clear. For vector primitives, there is an interesting analogy to the well known run length encoding (RLE) [1.5] used on unstructured images; if the vector is qualified by the 4-tuple $(x,y,\Delta x,\Delta y)$, then the number of bits required to specify $\Delta x,\Delta y$, increases as the logarithm of the vector length-- precisely the same variation obtained in RLE for runs along a raster line.

With increasing structure, communication efficiencies become striking. If a subpicture has been defined in a transmission or is already known to the receiver, then the next instance of it can be specified simply by giving the name and attributes. Minor modifications to existing images are also facilitated. With transmission of the redefinition of a subpicture, all windows could, for example, acquire curtains or become circular. A whole new house could be added just by naming "big house" and its colour, position, etc. attributes. It is important to note, by the way, that despite the use made of the line drawing Figure 1.2 for examples, the above comments about structure apply equally to filled area images such as Figure 1.1.

The exploitation of structure to reduce computational load, communication load and response time is implicit in the rest of this report.

1.2 ORGANIZATION OF THE REPORT

The point has been made that structured image communication is different from facsimile transmission. Economic areas where structured images are attractive or essential are outlined in Chapter 2. The remainder of the report is technical in nature, starting with a brief outline of standard remote computer graphics techniques in Chapter 3. Chapters 4 and 5 present surveys of graphics hardware and communication technology, respectively. Particular attention is given to important recent developments (e.g. semiconductor. technology, plasma panels, commercial packet switching) as they affect remote graphics. Chapter 6 attempts a synthesis of the many parallel technological changes with a discussion of the system organization required to exploit them; it is , in part, a look into the near future of graphics research and development. Finally, Chapter 7 summarizes the main findings of the report.

REFERENCES

[1.1] D.O.Hebb, The Organization of Behaviour, John Wiley,1949.

[1.2] Dean E. Wooldridge, The Machinery of the Brain, McGraw-Hill,1963.

[1.3] Tom M.Cornsweet, Visual Perception, Academic Press, New York and London, 1970.

[1.4] Donald P.Greenberg,"Computer Graphics in Architecture," Scientific American, May,1974.

[1.5] J.Capon, "A Probabilistic Model for Run-Length Coding of Pictures," IRE Trans. Inf. Th., vol.IT-9, pp.157-163, December, 1959.

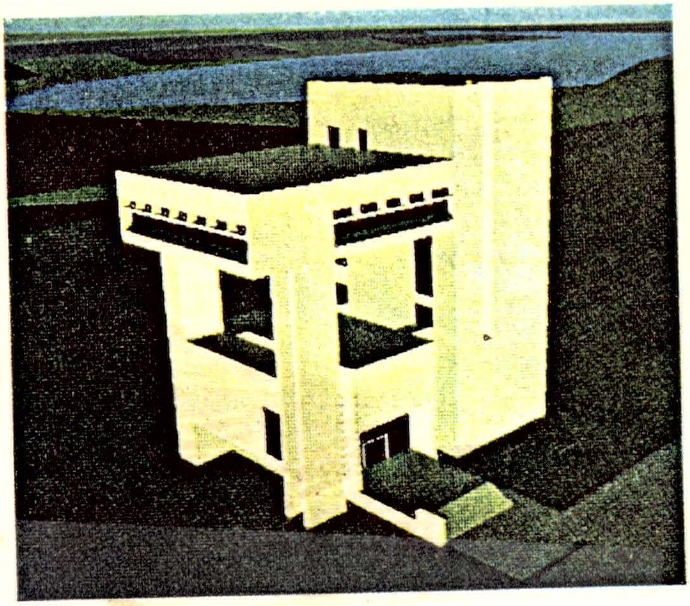


Figure 1.1 Filled Area Image

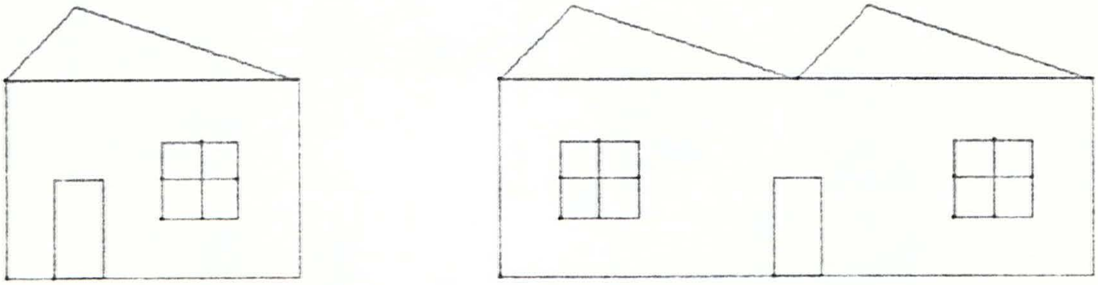


Figure 1.2 Line Drawing

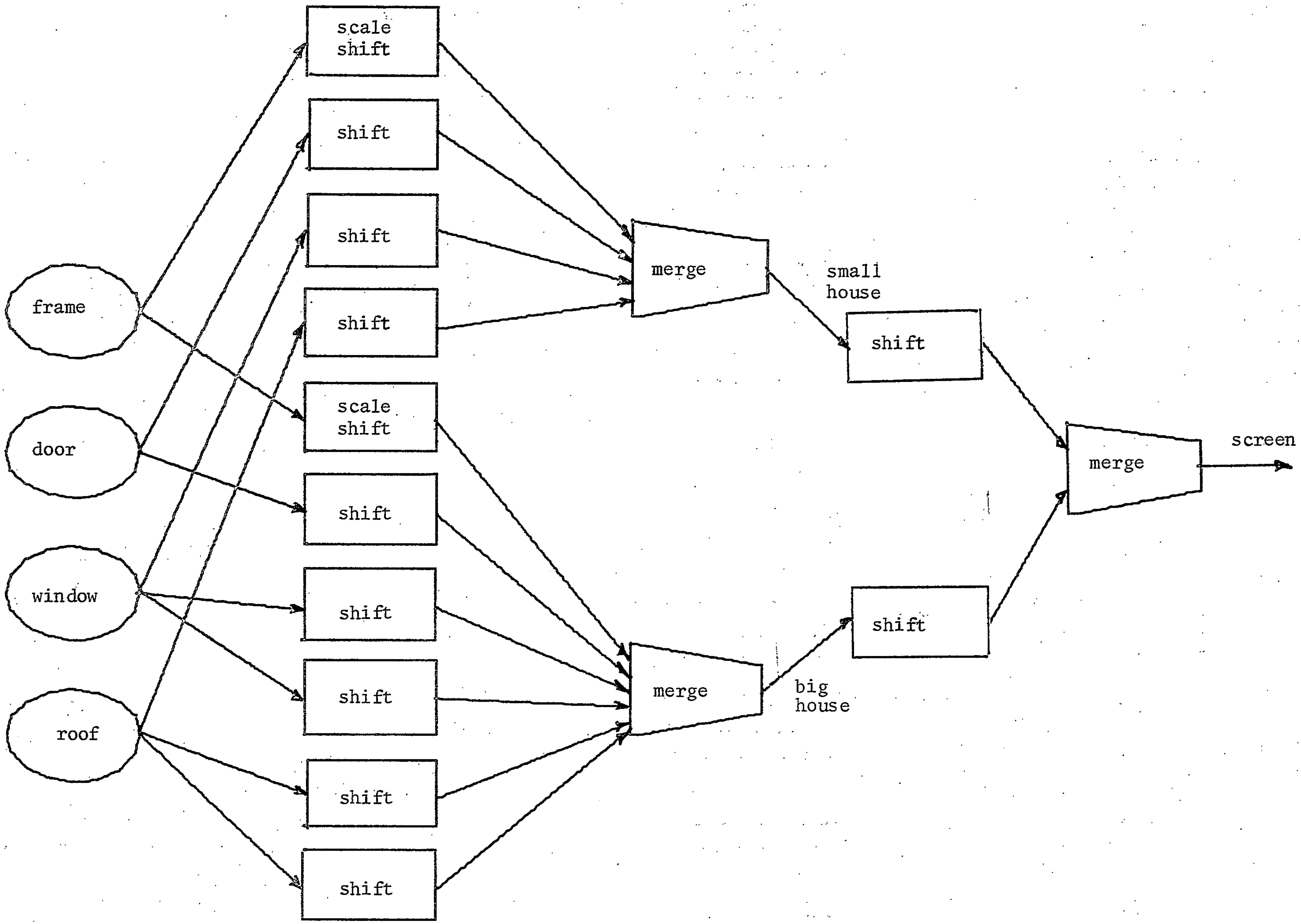


Figure 1-3 Structure of the Image of Figure 1.2

2. APPLICATIONS OF REMOTE GRAPHICS

The preceding chapter attempted to describe in general terms the type of graphics and computation covered in this report. Subsequent chapters will treat remote graphics also at a rather general level. Since it is important that the reader be able to relate the discussion to possible usage patterns, this chapter identifies six application areas Where possible, their description includes the current level of activity and future prospects for each. Many applications of computer graphics which appear to be impossible to implement remotely (such as display in a flight simulator for pilot training) have been ignored. The technical details will be abbreviated, since they are the subject of later chapters.

2.1 THE EDUCATIONAL TERMINAL

With the commercial advent of electronic computers in the early 1950's, educators recognized the potential of the computer for automated instruction. With the development of timesharing in the early 1960's (e.g. MIT's Project Mac), this potential approached economic feasibility. It was sufficient to encourage literally hundreds of programmers and teachers to create literally thousands of computer assisted instruction (CAI) packages -- most of them rigid, unresponsive and dull. These factors, together with the hyperbole which cloaked their efforts, gave CAI a notoriety which persists to this day.

A second wave of slower and more realistic experimentation began in the late 1960's and early 1970's. This work has produced both the University of Illinois-based PLATO system (which has not entirely avoided hyperbole) [2.1] and Ontario's CATS-OISE network [2.2], which had its inception at the National Research Council of Canada. It is now generally conceded that CAI has a cost-effective niche in the educational system. Primary areas are the teaching of skills, or rote learning, and instruction for the emotionally or intellectually handicapped [2.3]. Neither of these, incidentally, need be as drab as it sounds, since the technology involved is capable of impressive visual, tactile and textual interaction.

Graphics is an obviously important component of CAI. Humans absorb and process visual information at enormous rates compared to those possible by other senses; indeed, this is the most important lesson learned by the aspiring filmmaker [2.4].

Consequently, graphics should be regarded, not as an amusing but expensive frill, but as an integral and efficient part of lesson presentation.

The detailed requirements of an educational terminal are still very much a matter of research, but certain general requirements are clear. Low level interaction, such as acknowledgement of user input, must be very fast. Higher level interaction, such as an indication of whether or not the student's response to a question was correct, must also be rapid in order to maintain the feedback and student interest essential to learning. Lexical and semantic processing should be sophisticated so that the user is not forced to take part in an annoyingly rigid "conversation". If the user group cannot read, then special audio-visual or tactile input/output devices are necessary. Pictorial graphic display is often dictated by the subject matter itself. Finally, there is a requirement for low capital and operating cost (or more precisely, high cost-effectiveness) since education is rarely well supported financially.

The above criteria dictate considerable local intelligence for several reasons: first, communication costs are reduced with local treatment of basic interaction; second, the response time is correspondingly improved; finally, the cost of processing and memory is dropping more rapidly than the cost of communication. The two educational systems mentioned above evolved to meet these criteria.

With the encouraging reports of the cost-effectiveness of current efforts in the CAI field, one can foresee a gradual, but steady, increase in the use of educational terminals. It is the author's view that educational use will never make up a significant fraction of total communication traffic. Nevertheless, research into graphical and textual transmission in education should be funded, since it is in this area that the minimization of communication and processing cost is particularly important.

2.2 SCRIBBLEPHONE

Scribblephone is a term coined by Gordon Thompson of Bell Northern Research [2.5] to describe a graphical adjunct to telephony. People using this device would be free to exchange sketches, handwriting, and other handrawn graphics during the course of a conversation by means of a special stylus and display arangement. Maintenance of a common graphic space would allow collaboration between technically oriented people, managerial people, and others who like to talk with their hands.

Scribblephone could have interesting applications in education. Lectures on any subject for which this type of graphics is sufficient could be broadcast over radio or telephone lines. In comparison with the extravagance of television technology, this form of illustrated talk is modest in its use of bandwidth and is potentially so in terms of equipment cost (which is primarily LSI electronics). An even stronger contrast between the two technologies arises if one considers recording and distributing such lectures. Scribblephone lectures can be recorded on ordinary audio cassette and played back through an ordinary tape recorder to a receiving terminal. Conventional video recording technology, however, relies on elaborate electromechanical synchronization which holds little promise for further price drops. It is interesting that Bell Labs has developed a blackboard-sized "scribblephone" transmitting terminal which is being used for internal seminars.

Scribblephone has unique requirements. The display need not be dynamically alterable; an acceptable minimum capability treats the display as a pad of paper which gradually fills with figures or doodles and must occasionally be cleaned off. Transmission delay is not a critical parameter unless the "pointing" function (voice and stylus sychronization) is included. Finally, bandwidth compression is required to reduce the bit rate of the digitized coordinate stream to telephone line capacity or lower.

Electromechanical contrivances for transmitting stylus motion in analog form have been available for many years and purely mechanical arrangements date back at least a century, to the telautograph. More recently, research into the electronic and mathematical design of a digital scribblephone has taken place at Bell-Northern Research and (separately) at Carleton University.

Bell Northern Research has developed a prototype scribblephone which has considerable potential because of its use of LSI semiconductor technology. An ordinary video monitor is refreshed from an array of RAMs which contains a bit map of the screen image. The user "draws" on the brightened screen with a light pen (a light sensitive stylus) which causes a microprocessor to set the bit in RAM memory corresponding to the light pen position. The effect is that dots under the pen are darkened. An enhancement uses the microprocessor to connect these with straight lines. The processing of the graphics component for transmission is less sophisticated; the coding is both inefficient and information lossy, so that a separate 1200 baud telephone line is required for the graphics portion of the conversation.

The emphasis at Carleton University has been on the coordinate coding algorithms rather than on hardware development. It is the belief of the author (who has directed this effort) that scribblephone will fail to gain wide acceptance until it is as portable as the current briefcase terminals. This requires that voice and graphics share a single telephone line, which implies substantial bandwidth reduction in the graphics component. Some of the findings from this research are discussed in Chapter 6.

It is difficult to assess the potential of Scribblephone. Certainly the cost of the stylus and display must drop by an order of magnitude before such devices become economic. There is also a problem in introducing the product -- one would not buy a terminal unless he were assured that enough others existed to form a user group of some critical mass. Nevertheless the prospect of conducting a graphics-oriented telephone conversation is intuitively appealing to most people. If the technology and marketing problems are resolved, Scribblephone could become a major contributor to remote graphic communication.

2.3 PROCESS MONITORING

The term process monitoring, as used in this report, will have a special meaning. The state of a physical system or process is continually measured by a set of transducers. Both system and transducers may be geographically dispersed. The measurements may be partially processed at the transducer site, and are passed on to a central processing site where display of the system state takes place. It will be assumed that the display is primarily graphical.

Several examples of process monitoring systems exist. Ontario Hydro, at a central control office in Toronto, displays the current state of its power distribution grid. This description includes such variables as switch gear and operating generators. The Trans Canada Telephone System maintains a similar display of the current state of Dataroute at its network control centre in Ottawa. Other cases of process monitoring readily spring to mind. Senior engineers in Department of Public Works have discussed running the electrical, heating, and ventilation systems of a building as a unit, with the state displayed by a schematic diagram which is altered according to measurements of temperature, humidity, air flow rate and other quantities. A similar display could be established in a large chemical plant, steel mill, automobile plant or other complex industrial plant. Store-and-forward communication systems and road traffic control systems coulc use remote graphics for display of the current state of their queues.

Air traffic control is an example of process monitoring. Ministry of Transport can display, at a central site, enroute flight information such as range, azimuth, altitude, flight number, gathered from a number of geographically separated military radar installations. The data is packaged and transmitted over 50 Kbps leased lines to the central site and to other major sites for display.

It could be argued that process monitoring does not involve graphic transmission, since the information consists of parameters of the physical process which normally enter a data base on which the display package feeds. Nevertheless, according to the structured view of images in Chapter 1, the parameters are simply modifications to subpicture attributes and, in this situation, are all that needs to be transmitted. If the main purpose of communicating those parameters is display, then it is genuinely graphic transmission.

2.4 TECHNICAL AND MANAGERIAL APPLICATIONS

Technical and managerial applications of computer graphics have always been the most heavily supported and most active areas. Typically, highly competent individuals use computer generated displays in the design of new products and systems, in research, and in the making of management decisions. The graphic component of the computer output may be a convenient aid to rapid assimilation of presented data, or it may be intrinsic to the application itself, such as the problem of geographic location of data concentrators in a network. An indication of the utility of graphics in these areas is given by the following list of examples. It is by no means complete, but does serve to indicate some general features which will be discussed at the end of this section.

As an aid to research and development, graphics has been used for: specification and visualization of complex three dimensional surfaces in consumer product packaging , and in automobile and aircraft design; display of deformation of structures in civil engineering and the aircraft industry; textile design; interactive design of filters by presentation of root locus and Nyquist diagrams; development of control algorithms for multilegged vehicles; and for automated cartography. Graphics may also be used to guide the operation and display the results of network optimization programs in the design of communication and transportation systems, and to display the results of preprogrammed numerical routines such as the Statistical Package for the Social Sciences.

Graphics has been less widely accepted at the managerial level, but there are indications that its use is growing. Planning models based on graph theory (such as PERT and CPM) are now common. In performing these calculations one can edit the structure and parameters of the network in accordance with revised estimates as the project progresses. Graphical output allows display of the network itself, the critical path, the slack view, the Gantt chart, etc. Basic software packages for graphic display of network calculations are already available, and the American Department of Defense has assessed a remote graphics system dedicated to these functions [2.6].

A less well explored area, but also one of interest to those in management, is display for information systems [2.7]. Large data bases, such as social, demographic and geographic packages are difficult to access and to interpret, particularly when input and output are alphanumeric. A carefully tailored graphic front end to these would give the managerial user convenient access to the system and understandable display of the data. Pie graphs, bar graphs, histograms, phase plots and other common means of communicating numerical data would be prepared instantly on request from the user.

From the above discussion, some general features emerge. The examples given all provide hooks into some computational or data base package. Whether the package is canned or is written by the user is of little importance to the functions performed, which are primarily those of rendering large amounts of numerical data down to a visually comprehensible format.

The technical requirements of the remote graphics terminal can also be delineated, but with less confidence. First, the large application program or data base will reside in the host computer. This should limit the duties of the terminal to formatting of data and to display maintenance. Displays may be very detailed but need not be dynamically alterable, factors which suggest use of a storage type display. Finally, the pattern of communication is overwhelmingly one sided, with great quantities of data streaming from the host to the terminal with only occasional queries in the reverse direction. This suggests that, where possible, some form of bandwidth compression be used to avoid excessive communication delay.

2.5 GRAPHIC NETWORKS

A graphic network, as used here, will mean a set of two or more serially interconnected computational centers which have roughly equivalent processing power, though in general different functions. By now the concept of a resource sharing computational network is fairly well understood, though it is usually limited to two active nodes connected through a communication subnetwork. Distributed processing can be useful, for example, if several processing centers have evolved separately, or for other reasons must be geographically separated. Since virtual memory or unlimited disk space may not be available, interconnection of the centers could allow large programs and large data bases to be partitioned among them in order to achieve sharing of resources. A specifically graphic network is a new concept, however, and needs a great deal of exploration before it can be evaluated.

At one level, a graphics network could be simply a point-to-point connection made through an existing network in order to access specialized hardware (e.g. a microfilm plotter) or software (e.g. a CPM analysis package). This approach was adopted in the ARPA network by Newman and Thomas [2.8], who developed a graphics message protocol as a superstructure to the packet switching protocol.

Another level of graphic network involves two or more Scribblephone nodes in a symmetric arrangement. One can easily visualize the graphic equivalent of a conference call and, perhaps less easily, imagine extended functions of such a node.

One such extension is the "common graphic work space," being developed by Bown and Sawchuk [2,9]. The participants at each node will be able to exchange graphic data, including sketches, as well as to access data base, numerical and graphical packages locally or remotely. The graphic conversation will be backed up by the full power of a computing system.

What sevices should be provided? An obvious possibility is a local package to aid sketching (horizontal and vertical lines, for example) or to allow quick computations (e.g.an APL interpreter). Another is a simple graphic data base analysis package, which would answer queries such as the number of instances of some subpicture (e.g. a NAND gate). Other services are less clear, but will no doubt evolve as experience is gained with this unique facility.

2.6 ENTERTAINMENT

The home entertainment market has seen a new entry in the last two years -- the video game. Typical of these are ping pong, hockey, and handball; the players use potentiometers to control the position of simple shapes on the screen of their home television set which is disconnected from the cable or antenna. It is a substantial market. With units priced in the approximate range $50 to $150, and expected sales in 1977 of over 500,000 units [2.10], the market is at least $25 million.

The video game is not simply an innocuous piece of gadgetry; it has rather extensive implications. Most important is the fact that attitudes toward the television screen are being irreversibly altered. No longer are viewers passively accepting or selecting material presented to them by the networks. With the game, the screen reacts to the viewer, as well as the viewer to the screen. A conciousness of new roles for the television set has been awakened.

What lies ahead? More elaborate games are one obvious possibility. But non-trivial games such as chess, backgammon, or war games require at least mini or midi computer intelligence. Educational units are another possibility, providing practice at numbers, letters, eye-hand coordination, etc. But again, limited device intelligence means limited utility. The real extension of the game may well be the addition of an acoustic coupler. If the home device is a simple graphics terminal hooked to a large timeshared computer through the local telephone network, the utility of a single device is increased by orders of magnitude.

If the home computer graphics terminal sounds far-fetched, consider the following. A video graphics terminal can be assembled for $1000 [2.11]. With mass production, falling semiconductor prices, and some simplification, the price could be only a few hundred dollars, less than many of the colour television sets and stereo systems purchased routinely in the home electronics market. As for the computer, the esoterica associated with job control languages can easily be hidden from the home user -- logging on and off, for example, can be performed automatically by his terminal. Well designed software packages can supply endless varieties of game: space war, backgammon, blackjack, etc. That is only the beginning. Educational packages with real interactive power, information searches, house or boat design, even novels and message drops -- electronic mail -- are all waiting to enter the home through the same device, the video graphics terminal.

There is a good deal of irony in the situation. For years, these futuristic services have been expected from the cable television companies. But while the Canadian Cable Television Association and CRTC have been wrangling over the development of the facility, technology has performed an end run around them. The services foreseen by the author involve no major changes to existing methods of remote computing through the telephone network. All that will change is the amount of this activity (potentially very large) and the people doing it (everyone).

What of the telephone companies? The home graphics terminal presents a real danger. Their facilities have been designed on the basis of an average phone call lasting about 5 or 10 minutes. Calls to computers are quite different, characterized by long holding times of half an hour or more. Some apprehension about this was voiced in the 60s and early 70s, but the level of data traffic was so low that the effects were not significant. Suddenly we now face a possibility of a very large number of data calls. Will the telephone plant be adequate to meet the demand? Even worse, from the point of view of the companies, there will be no extra revenue for them (apart from a temporary increase in families requesting a second phone line) since all the calls will be local.

There are clearly a number of unsettling issues raised by the simple video game.

REFERENCES

[2.1] D.L.Bitzer, "PLATO: A Computer-Based System Used in the Engineering of Education," Proc. IEEE, vol.59, no.6, pp.960-968, June 1971.

[2.2] Documents available from Clarence Payne, Director, Computer Center, Algonquin College.

[2.3] R.M.Knights and D.H.Richardson, "Automated Assessment and Training of Retarded and Disadvantaged Children," Research Bulletin no.10, Dept. Psychology, Carleton University, Ottawa, February 15, 1974.

[2.4] R.P.Madsen, The Impact of Film: How Ideas are Communicated Through Cinema and Television, MacMillan, New York, 1973.

[2.5] G.Thompson and M.Westelman, "Scribblephone -- Extending Man's Powers of Communication," Telesis, pp.74-79, September 1968.

[2.6] J.Potocznlak et al, "Affordability of Computer Graphics for Planning Networks in DOD Program Management," Computer Graphics (SIGGRAPH-ACM), vol.10, no2, pp.212-217, Summer 1976.

[2.7] P.P.Tanner, "Dynamic Graphical Display of Data," Proc. 4th Man-Computer Communications Conf., pp.22.1-22.7, Ottawa, May 26-27, 1975.

[2.8] R.F.Sproull and E.L.Thomas, "A Network Graphics Protocol," Computer Graphics (SIGGRAPH-ACM), vol.8, no.3, pp.27-51, Fall 1974.

[2.9] H.G.Bown and W.Sawchuck, "An Interactive Visual

Communication System," submitted to Computers and Graphics, 1976.

[2.10] Ron Blunn, "Video Games a Profitable but Risky Business," Ottawa Citizen, pg.12, Tues, Jan 11, 1977.

[2.11] F.Baskett and L.Shustek, "The Design of a Low Cost Video Graphics Terminal," Computer Graphics (SIGGRAPH-ACM), vol.10, no.2, pp.235-240, Summer 1976.

3. REMOTE DISPLAY TECHNIQUES

This chapter is included as preparation for the following chapters. The basic functional components of a graphics system are introduced here, and are described with an eye to the communication boundary which separates them. As the intelligence of a graphic terminal is increased, it will acquire more of the functions and the boundary will shift. Three natural stopping points in this evolution are described.

3.1 GENERAL SYSTEM CHARACTERISTICS

Graphics in the 50s and 60s developed rapidly. The natural consequence was a multiplicity of techniques and special hardware, many of these known by several different names. These diverse approaches were formalized in 1972 by Newman and Sproull [3.1]. This book established a common vocabulary and description, and remains the key text for workers in the area. A slightly modified formulation was presented again in 1974 [3.2] and this discussion is adapted and extended from that work.

Figure 3.1 shows the various functionally defined modules of a general purpose graphics system. In any particular implementation, of course, some of these modules may be coalesced or absent entirely. Of those present, some may be located at the host and others at the terminal; further, many may be implemented in either hardware or software, though this is immaterial to the present discussion.

Before the description of each module, let us consider what the overall system can do. At the user's request the results of computation, measurement, or data retrieval can be displayed in some form on the screen. A structure of subpictures (Chapter 1) is maintained, so that entities can be added, altered, or deleted without affecting the entire display. The user can direct the flow of computation and the generation of data structures by the conventional teleprinter or by special purpose graphic input devices. Examples of these include the light pen to select entities from the screen, and the stylus and digitizing tablet to sketch a picture. Other than the extensive use of geometry and the specialized input devices, there is little to distinguish graphics from other forms of computation.

To begin the discussion of Figure 3.1, note that there are significant non-graphic components shown. The general purpose data base and non-graphic procedures are included for completeness. Typical of these are census or geological data or a maxflow analysis package.

The application program is written by or for the user and harnesses the services of the computational and graphics facilities. It acts on user input and performs most of the picture definition computations. Normally written in some high level language, it is provided with hooks into the graphics facility by the graphics package.

The first significant data structure after those internal to the application program is the structured picture definition. This is a representation of the image which is displayed, in whole or in part, on the screen. It is structured to allow separate subpictures with different attributes. These subpictures are often defined in their own coordinate system with real numbers (e.g. in kilometers). They may also be fundamentally different in nature from the screen image if solid colour images are displayed. The attributes can consist of a transformation matrix to bring the picture definition into correspondence with the viewpoint coordinate system used for the screen, windowing information, and colour, intensity, and texture information, as well as a name. Data flowing from the application program to build a structured picture definition is in a fairly flexible format, which provides an opportunity for data compression, but includes all of every subpicture, visible or not, so it involves a large amount of data. Updates are easy, since position, orientation, colour, etc. Are attributes and the whole subpicture need not be redefined. And even if a subpicture must be redefined, the rest of the structured picture definition may be left intact.

The segmented display file is constructed from the structured picture definition by the transformation, clipping, and visibility routines. These convert the subpicture from its original coordinate system into that of the screen by scaling, rotation and translation, then modify the subpicture if it falls partially or totally outside the screen (or viewport within the screen) boundaries. Next the various subpictures of a three-dimensional scene may be compared to determine if portions of one are obscured by another one closer to the viewer. The resulting segmented display file is usually a linear string of two-dimensional display primitives, expressed in integers, with a subpicture structure. The attributes are rather meagre compared with those of the structured picture definition, usually consisting only of those which the available hardware can interpret, such as name, intensity/colour, texture, light pen sensitivity or blink enable. Changes to these attributes and the deletion of subpictures can be done with minimal data flow. Modifications to, for example, position or orientation are not generally possible at this level since the data structure is two-dimensional and clipped, so these require redefinition of some or all of the subpictures. Addition or redefinition of subpictures takes more data, of course, but still does not require rebuilding the entire display file. Building the segmented display file after the clipping and visibility routines normally involves a lower communication load than does building the structured picture definition, because of the potential elimination of large parts of the scene which are hidden or off screen.

Although the segmented display file is often interpreted directly by a hardware display processing unit to drive the screen, especially in the case of vector-type displays, recent technological changes make it worth considering an extra stage of memory. Display memory is the raw image in unstructured format (Chapter 1) and consists of semiconductor random access memory. A very high data flow is involved in building this, since the device which interprets the primitives of the display file outputs a lot of dots!

Finally, the display driver converts the bit patterns given to it from the display memory or from the display file interpreter (if there is no display memory) to voltages driving the beam position and intensity. For devices which incorporate their own memory (e.g. the storage tube), this process need be done only once.

The missing element in the picture is the user who views the image, then provides the application program with input from any of a number of devices, some familiar and others unique to graphics. These signals are formatted appropriately by the input routines to close the loop. The information rate of these devices is a trickle compared with the flow in the forward direction.

3.2 THREE LEVELS OF INTELLIGENCE

The previous section described the functional organization of a graphics sysem. Since this report is concerned with remote graphics, the host and the display are assumed to be geographically separated. There is a communication boundary separating the two, across which information flows bit-serially. Selection of the communication boundary -- the location of the various procedures and data structures -- is governed by two main factors.

The first factor is the improvement due to parallel processing. Smart terminals are often described as off-loading functions from the host. This is really a way of achieving speed with dual processing. As the communication boundary is pushed far back toward or into the application program, the performance actually begins to degrade. The terminal -- now an intelligent satellite -- does the bulk of the work, and some dual processing gains become lost. Some of the terminal's duties should be off-loaded back onto the host, especially since computation is likely to be cheaper on a large time-shared machine anyway.

The other factor affecting the choice of boundary is the nature and extent of data flow across it. Points of high data flow are clearly unsuitable because of the communication cost and delay, which would offset any gains in speed due to parallel processing. Points where data is in a rigid format are also slightly suspect because of the limited scope for data compression tricks. An example is the input to the segmented display file, which consists of a list of hardware interpretable display primitives.

In recent literature there have been several attempts at standardizing display commands [3.4-3.5]. This virtualization of display devices is intended to allow graphics software to become relatively device independent. A standard set of primitives, for example, is an excellent idea. Higher levels of virtualization have been described with a standard set of segmented display file manipulation commands. Norpak Inc. has in fact produced an intelligent graphics terminal which interprets a set of such commands called GTIs [3.5]. In the light of the above discussion one can see virtualization as formalization of the communication boundary location as well as the formats to be used.

Here we examine three representative partitions of intelligence between host and satellite. No claim is made that the three levels are definitive. Many variations are possible, but the three to be discussed are natural stopping points in the evolution of the remote graphics terminal.

3.2.1 The Remote Plotter

This configuration, shown in Figure 3.4, allots very little intelligence to the terminal, which acts essentially as a remote plotter by interpreting primitives. It is not required to treat any part of the display as a separately distinguishable entity. For this reason, the display file, display processor, and display shown separately in Figure 3.1

are often combined into a single unit. Examples are the Tektronix 4010-4015 series of storage tube displays, the Data Disc refreshed video displays, and the Norpak RG-4000 semiconductor refreshed video display (and, of course, the x-y plotter itself, which is adequate for some purposes).

The unstructured display does not imply restriction to line drawings of low complexity; indeed multicoloured, shaded, three dimensional scenes are allowable if the display hardware supports this. The visibility, clipping and shading algorithms, however, would have to be performed on the host machine, since they depend on a structured scene description.

The communication load imposed by this configuration is very high. Flowing from host to terminal for a vector display is a string of coordinate pairs of from 16 to 26 bits each (depending on the display device), and images of 1000- 10000 vectors are not uncommon. The resulting delay precludes effective graphic interaction in the usual sense, as anyone who has tried 300 baud (or even 2400 baud) graphics will confirm. User input is therefore limited to characters, such as a request for another plot, and positional information, such as crosshair location. A very asymmetric data flow is characteristic of this mode of operation. The significant exception to the above comments regarding interaction and asymmetry is the scribblephone. Users exchange low rate streams of coordinate pairs generated by stylus and tablet or light pen and raster-mode screen. The displays at the two ends are identical and are unstructured, even though both users have contributed vectors.

3.2.2 The Display File Processor

This level of remote intelligence requires the terminal to construct and interpret segmented display files. The files are organized as a collection of individually defined entities each with its own attributes such as name, intensity/colour, texture, on/off status, light pen sensitivity and position. "Light buttons" in a menu or logic symbols in a circuit diagram are both examples of such entities. Some systems allow subentities which inherit the attributes of the parent entity, except for additional naming information. Another useful feature allows an entity to be defined once, but used several times (with different attributes) as a "graphic subroutine" (Chapter 1).

The attributes and coordinate or character information of each entity are normally interpreted by a special purpose display processor to regenerate the display. This may have to be done several times per second in the case of a refreshed CRT. Conversely, for a storage display, it need be performed only once after each modification of the display file. This latter case has the disadvantages of low speed and awkward light pen interaction, but the advantage that responsibilities of the display processor can largely be assumed by a general purpose processor at the terminal.

The duties of the terminal's processor are reasonably well defined. In addition to communication processing, it must respond to commands received from the host computer to add or delete entities in the display file or change their attributes, where the entities are referenced by name. It must also intercept raw interrupts from user devices and package these appropriately for a return message to the host computer, which can deal with them asynchronously. A light pen strike or stylus interrupt, for example, could result in the transmission of the x y coordinates or simply the name of the graphic entity, where the terminal's processor performs any calculations required to determine the name. Yet another processor function is memory management; the display file can be arranged as a linked list of fixed size blocks to avoid memory fragmentation when entities are deleted.

It is evident that the communication load is much reduced from the case considered earlier, the remote plotter. Definition of the graphic entities still requires transmission of a string of coordinates from the host to the terminal, but after this phase minor modifications to the display can be performed with short commands, and the display is regenerated locally. This has two effects: the communication cost is reduced and the response time is reduced. It should be remembered however, that this performance improvement is bought with increased terminal complexity and expense.

A terminal with this level of intelligence is appropriate for most of today's interactive graphics work, and in particular to the remote educational terminal, with its requirement of low response time to simple user demands.

3.2.3 The Programmable Terminal

As the intelligence of the terminal is increased still further, we reach the level of a structured picture processor. The terminal now incorporates the structured picture definition of Figure 3.1 and the transformation, clipping, and visibility procedures. To avoid massive memory requirements, the segmented display file is simply eliminated, and the picture is drawn directly onto a remote plotter type of display.

The key element in the definition of the structured picture processor is the inclusion of the transformation, clipping, and visibility procedures, which represent perhaps the greatest computational load in the entire system. This may not be particularly smart if they are to be performed in software, since the host would undoubtedly have a superior floating point processor. However special purpose terminal hardware for these routines would speed up the process enormously. One pays for these luxuries, of course.

As noted earlier, this is a convenient point to locate the communication boundary, since the flexibility of the data format lends itself to data compression, and since many modifications to the display can be made simply by changing the attributes, now rather extensive, of a subpicture.

REFERENCES

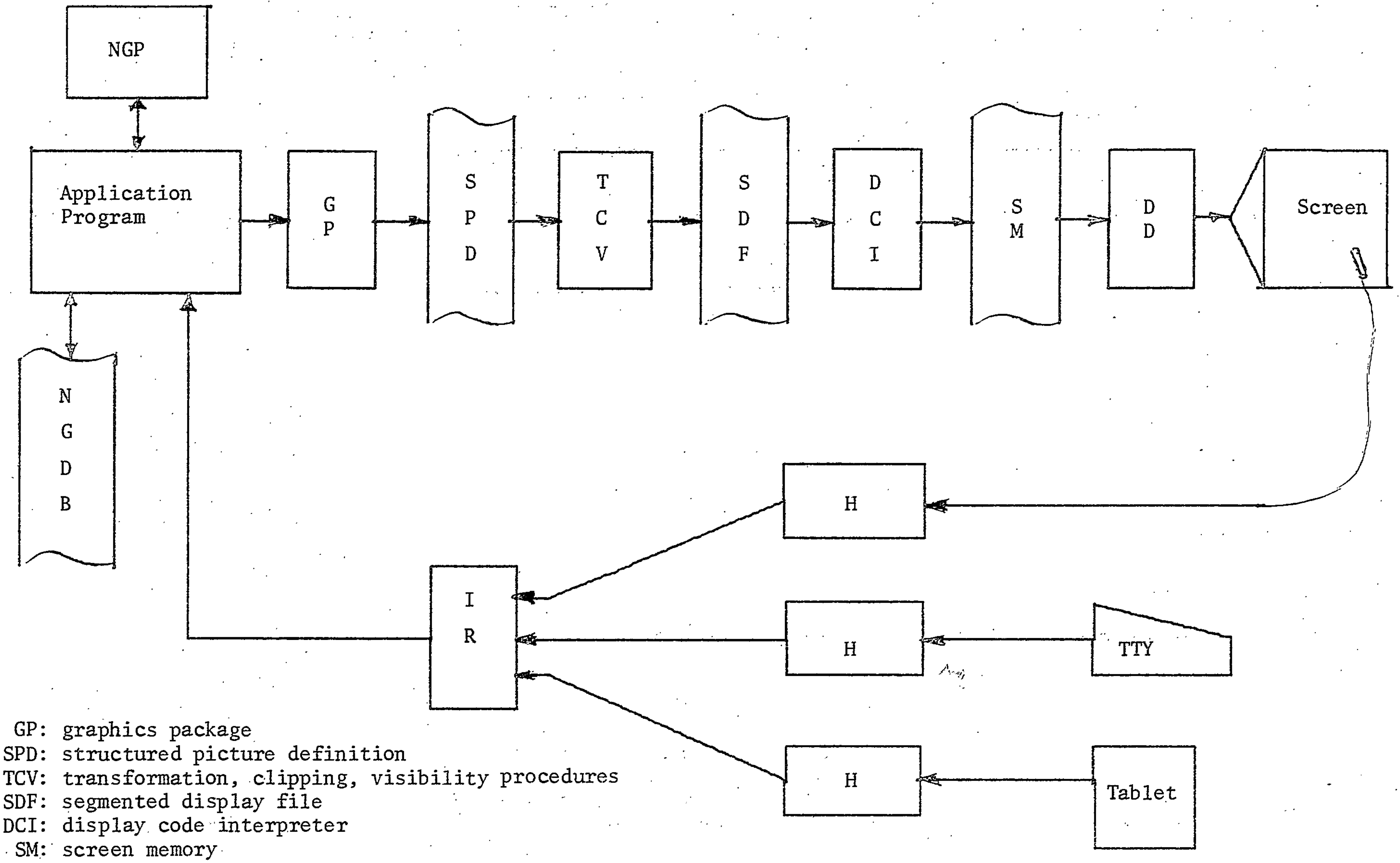
[3.1] W.M.Newman and R.F.Sproull, Principles of Interactive Computer Graphics, McGraw-Hill, 1973.

[3.2] R.F.Sproull and E.L.Thomas, "A Network Graphics Protocol," Computer Graphics (SIGGRAPH-ACM), Vol.8, no.1, pp.27-51, Fall 1974.

[3.3] W.M.Newman and R.F.Sproull, "An Approach to Graphics System Design," Proc. IEEE, vol.62, no.4, pp.471-483, April 1974.

[3.4] Jack R.Davis, "Device Independent Graphics Software -- Is It Possible?" Computer Graphics (SIGGRAPH-ACM), vol.9, no.1, pp.232-245, Spring 1975.

[3.5] H.G.Bown et al,"System Independence for Interactive Computer Graphics Application Programs," Proc 4th Man-Computer Communications Conf., pp.1.1-1.12, Ottawa, May 26-27, 1975.



GP: graphics package
SPD: structured picture definition
TCV: transformation, clipping, visibility procedures
SDF: segmented display file
DCI: display code interpreter
SM: screen memory
DD: display driver
H: handler
IR: input routines
NGDB: non graphic data base
NGP: non graphic procedures

Figure 3.1 Functional Organization of General Purpose Graphics System

4 GRAPHIC TERMINAL HARDWARE

The previous chapters have ranged from general graphic considerations, through application areas, to the functional organization of the graphic terminal. To continue this trend to the specific, the present chapter concerns itself with the hardware technology which underlies the functional description. This discussion is important for two reasons. First, some of the display and memory devices can incorporate more than one function, a fact which has a direct bearing on the cost versus level of intelligence tradeoff. Second, the outlook in the semiconductor area has been and will be changing rapidly as new technologies are introduced and old ones drop in price. The effect will be felt primarily as an increase in local intelligence and capacity, which will allow present-day experimental techniques to be implemented routinely.

4.1 COMMUNICATION HARDWARE

The communication functions of a terminal lie between the signals used on the communication line and the messages presented to and generated by the terminal's processor. Within this range, two layers can usefully be distinguished. The first layer is the conversion between channel waveforms and "raw bits" (internal bits, accessible by the processor, but possibly unformatted and possibly in error due to line noise). The second layer is the conversion between raw bits and error-free messages acceptable by the local operating system, if any. This second layer is primarily concerned with communication protocols, and is not commonly implemented in graphic terminals at the date of this writing.

Within the first layer are one or more blocks. A modem is required if telephone line transmission, rather than a direct baseband connection, is used. Low speed modems often incorporate IC digital phase lock loops for detection. The terminal side of the modem employs a serial synchronous or asynchronous stream of bits or characters [4.1]. The second block is a UART (universal asynchronous receiver/transmitter) or USART (.... Synchronous/asynchronous...). One of the early triumphs of LSI techniques, this chip performs character sync, serial-parallel conversion, simple parity checking, and presents status flags all for the present day price of about $15 for the basic receiver/transmitter chip. Certain enhanced versions (such as Digital Equipment Corporation's DL-11 and DU-11) allow control of the ring and answer functions of a data set.

Another useful LSI chip in the first layer, particularly for asynchronous communication, is the FIFO character buffer, which provides a parallel in-parallel out queue. . Critical processor response times are eased by this automatic buffering. The second layer, as noted above, is much less a matter of connecting off-the-shelf items. However DEC has already brought out a synchronous line adaptor (DUP-11) which incorporates the UART functions as well as automatic sync generation, bit stuffing (as required in SDLC and HDLC bit oriented protocols) and error checking (by means of a standard CRC-16 polynomial). The three latter functions were formerly implemented with software, so that it represents a significant off-loading of the processor. When other manufacturers bring out their own versions the price will no doubt drop. There seems little reason why the remaining functions of most protocols (primarily message sequencing and retransmission control) could not be microprogrammed. One can expect to be able soon to purchase hardware implementations of common HDLC or X.25 communication protocols. DEC has already announced its DMC-11 [4.2] to implement its DDCMP line protocol. In effect, specialized multiprocessing would be employed within the terminal itself.

4.2 MEMORY

The discussion of Chapter 3 showed that memory had three main functional roles in the graphics terminal. These were program storage, data base storage and display file storage. This section will provide a brief discussion of the relevant technologies with cost estimates where this is possible.

4.2.1 Core

Core needs little discussion. It is versatile--suitable for all three memory functions. It is non-volatile, so that power interruptions leave the contents unaltered. Unfortunately, in comparison with certain semiconductor technologies it is slow (barely submicrosecond cycletime), bulky, and expensive (about $2500 for a module of 16K-16 bit words.).

4.2.2 Semiconductor RAMs and ROMs

Random access memory, in the form of LSI arrays of bistable memory elements, is a direct competitor of core memory for data base and display file storage. It is faster, more compact, and will soon be cheaper than core memory. Its drawbacks are volatility (information is lost on power failure) and temperature sensitivity.

Price and availability of RAMs are changing rapidly, but the present picture is about 0.5 cents/bit for 1K bit static RAMs and slightly less for the 4K bit chip (available as dynamic RAM only). The 16K bit RAM is not yet on the market.

but predictions [4.3] suggest that upon its entry the price will drop rapidly from the initial value of about 0.1 cents/bit.

Concerning use of RAMs for display file, two modes are possible. In the first, a specialized display processor cycles through the file, interpreting commands and coordinate values to create the screen image (usually vectors, dots or characters). In a sense the screen is randomly addressed, since any coordinate pair can be addressed in any order. This is the mode used for most dynamically alterable, interactive displays, including those using core memory. In the other mode, the display file contains a pixel-by-pixel version of the screen image which is read out in raster fashion to a video monitor. Display file size and screen flicker do not depend on image complexity, and surfaces can be displayed as easily as lines. It has the disadvantage that it is difficult to implement selection of subimages, since most structure has been destroyed.

At current prices, raster mode display memory for a 512 x 512 pixel, black-white image would be approximately $1250, exclusive of decoding and addressing functions. Gray levels or multiple colour selection can be added incrementally with more bits/pixel. With a 16 Kbit RAM the above figure would be less than $250. This prediction has encouraged manufacturers, notably RAMTEK, Hughes Aircraft, and Norpak (of Pakenham Ontario), to market raster displays using semiconductor memory. BNR's prototype Scribblephone uses the same technique. An interesting recent entry from Matrox Electronic Systems of

Montreal is a 256x256 black/white display unit intended to buckle onto an Intel 8080 microprocessor. It provides point write/erase primitives and standard video output for $630.

ROMs are also required in a terminal, even of the remote plotter variety, unless the user is willing to load the control program from paper tape at each session. (Downline loading from the host is possible, but still requires ROM-based communication handlers). ROMs are somewhat less expensive per bit than RAMs and are available in up to 16K bit chips.

4.2.3 Magnetic Bubble Memories

Magnetic bubble memories [4.4-4.5] are less familiar than core or N-MOS RAMs. A simple repetitive pattern of permalloy bars is deposited on a garnet epitaxial layer. Under the influence of a magnetic field rotating in the plane of the substrate the basic memory elements --"bubble" shaped magnetized domains-- are induced to move from one bar to the next. MBMs therefore consist of recirculating shift registers, with the presence or absence of a bubble as the two possible memory states at each location.

Among the virtues of these devices are their high capacity low power dissipation and simplicity of manufacture [4.5]. They are non-volatile, in contrast to the faster charge coupled device memories. The price is also right-- as of this writing the first commercial MBM has just been announced by Texas Instruments [4.6] as a 92 Kbit chip costing 40-50 millicents per bit !

With regard to their possible use in graphics terminals, one should regard MBMs as small high-speed disk equivalents. The announced average "rotational latency" to access a bit is 4 msec, and the transfer rate is 50 Kbits/sec. Replication on a 1 bit/chip basis would give 50 K words/sec. An intriguing possibility is storage of the display file in bubble memory in a sequential format with no display jumps. Instead of the display processor cycling through memory, the display file would be cycled past the display processor. Very large display files with a sequentially accessed structure would be economically feasible.

4.2.4 Charge Coupled Devices

CCD's are, like magnetic bubbles, a form of recirculating shift register memory [4.7-4.8]. They are, however, based on a mechanism for transferring packets of charge from slot to slot in the device and can therefore act as analog shift registers as well.

CCD memories in 16 Kbit chips are on the market. Compared with bubble memories they are faster, with reduced latency (about 4 msec average) and increased transfer rate (up to 2 Mbits/sec for the Intel 2416)--but are volatile. They are somewhat more expensive than bubbles. The Intel 2416 costs $35, about 225 millicents per bit.

They are suitable for slow scratch memory or very high speed disk equivalent. Their principal application wil probably be as raster mode screen memory, since they are fast and sequentially organized. There are difficulties in writing vectors into a strictly sequential memory, but they have been solved in other areas such as disk-refreshed graphics. In the graphics context, CCD's may prove useful as storage for sequential display files as described for magnetic bubbles.

One fascinating possibility in the use of CCD's for graphics relates more to processing than to storage. Discrete time analog finite impulse response filters can be constructed in which the tap weights are processor settable [4.9]. The obvious application is analog two dimensional transformations of vectors before they are drawn on the CRT. This "instantaneous" transform would off-load a very large chore from the processor.

4.2.5 Disk Memory

In disk refreshed displays, a standard video monitor is driven by a specialized video disk on which is stored the screen image in raster format. Current models are bit addressable for write and erase functions and are compatible with standard video studio, recording and broadcast techniques. The main virtue is the cost sharing provided by multi-track multi-head disks-- up to 80 independent channels are available on the Data Disc 5400 series. The principal disadvantage in the context of remote graphics is that the disc is normally cosituated with the host computer and distribution of the signal is by expensive broadband video facilities.

4.3 DISPLAY DEVICES

The display itself is likely to prove the limiting factor in the general downward trend in prices for most graphic terminals. The semiconductor components of the terminal take advantage of LSI technology, but the display is a precision piece of glass/metal manufacturing and (usually) analog electronics. Until a semiconductor display element is capable of being replicated in large quantities (512x512 or 1024x1024) to form a flicker free display, graphic terminals will be limited by the display.

The one exception to the situation is the appearance of terminals with memory sufficient to drive a standard 525-line video monitor, as discussed below.

4.3.1 Refreshed CRT (Vector and Raster)

The refreshed cathode ray tube uses a screen phosphor with a decay constant on the order of 50 msec. The electron beam must continually retrace the image if it is not to disappear (and retrace faster than about 30 times/sec if it is not to flicker). It is therefore suitable for dynamic graphics in which the image, or portions of it, must be altered rapidly in response to user inputs or computed changes. Two modes are commonly used4 vector mode and raster mode.

Vector (cursive) mode allows the intensity and x and y deflection signals to be varied in arbitrary patterns so that the beam traces straight lines, dots or characters on the screen. In the case of the Hughes Conographic display processor, special analog circuitry allows conic section arcs to be generated. The basic deflection signals can be produced by analog integrators or digital counters operating digital to analog converters, but the drive circuitry is analog, which accounts for the linearity, offset and overshoot problems associated with vector mode CRTs.

From the basic strokes, complex line drawings with hierarchical tree or lattice structures can be displayed. The structuring of the image allows easy selection of screen entities by light pen, and dynamic modification of their attributes.

Raster mode of course, refers to the television type of display in which the beam sweeps the CRT in a predetermined line-by-line fashion. The two commonly available resolutions are the 525-line North American television industry standard and a high resolution 1029 variant. (In Europe, 1229 lines is standard resolution.)

The simplicity of this format is bought at the expense of large screen memories and the lack of structure in the image. The latter leads to difficulties in implementing the "pick" or entity selection function, although a partial solution has been implemented in Norpak's RGS series.

Despite these problems, the ubiquity and simplicity of the standard video monitor, coupled with the falling cost of semiconductor RAMs for screen memory make this an increasingly attractive configuration for the inexpensive graphics terminal of the future.

4.3.2 The Storage Tube

The storage tube is a cathode ray tube in the sense that x and y deflection signals drag a beam and the corresponding strokes appear on a phosphor-coated screen. The beam, however, writes onto a dielectric-coated grid immediately behind the screen which retains the charge transferred to it by the beam. A separately generated low energy "flood" of electrons illuminates the grid and is repelled by the negatively charged regions. At the positively charged areas the flood electrons pass through the holes in the grid and strike the screen to produce the visible pattern of lines.

The main advantage of the storage tube is that a separate display memory is not required, since written images persist for hours. This factor results in a very simple graphics terminal, such as the Tektronix 4010 series, which consists basically of UART, command interpretation and vector generation hardware, and a storage tube.

On the other hand, the storage tube has limited application outside of the remote plotter role. Not only has the original picture structure been destroyed, but there is no selective erase feature. Modifications to the image require that the whole image be redrawn with corresponding long communication delays. Recent work [4.10-4.11] has used local processing and storage to maintain a local data base, from which the image is regenerated after changes. The delay is much smaller, but is still noticeable.

4.3.3 The Plasma Panel

The plasma panel is the newest entry in the field of display devices. It was developed at the University of Illinois and is now marketed by Owens-Illinois. Basically it consists of two closely-spaced glass plates with the intervening space evacuated and back-filled with neon. The plates each have a closely spaced linear grid of parallel electrodes, with one grid horizontal and the other vertical, and an AC voltage is applied between the two grids. If the voltage at the intersection of an x wire and a y wire is raised momentarily, the gas breaks down and discharges, creating a glowing orange point. As usual with gas discharges, it is a bistable phenomenon and requires a momentary lowering of the voltage to extinguish the glow.

The result is a display which, like the storage tube, forms its own intrinsic screen memory but is point or rectangle addressable for write and erase. The selective erase feature gives it potential for dynamic displays, although there are minor problems. Because it is transparent it is readily superimposed over background detail, such as business forms or photographic slides.

At the moment the manufacturer has provided supporting logic for the alphanumeric set, but not for vectors. The controlling device must therefore perform a series of single dot writes. Although the write itself takes only 2 usec, the arithmetic computations associated with line drawing are very slow when performed in software, particularly by a cheap microprocessor.

The plasma panel is used in the PLATO CAI system, and a plasma panel based educational terminal has been developed by the National Research Council [4.12]. Small alphanumeric plasma displays are used in business terminals.

Perhaps the most exciting recent development is the announcement by Bell Labs [4.18] of a simple method for detecting the position of a light pen over a plasma panel. It is claimed that the additional hardware required is negligible. If this is the case, then the future of the plasma panel in interactive graphics is assured.

At the moment, the price (about $3500) limits the spread of plasma panels in the display industry. Although there is no indication of a future drop in price, perhaps the addition of light pen interaction and hardware vector capability will make this device more attractive.

4.4 USER INPUT DEVICES

User input devices provide the interaction in interactive graphics [4.13]. For the most part, they are a relatively inexpensive component of the graphics terminal but, since they are unlikely to benefit from forseeable technological changes, they may ultimately act as the limiting factor (after the display itself) on the expected price drop in simple terminals. They have been included in the report for this reason, and because they have attracted some very imaginative effort-- they are fun to build and use!

As noted in Chapter 3 the great variety of such devices reduces to a very few functional roles. It can also be shown [4.14] that these roles are logically equivalent--for example, light pen selection of letters from a displayed alphabet is logically equivalent to a keyboard--though it does not follow that they are equally convenient. This report adopts the following classification of functions: pick-a selection or identification of a subentity on the screen; push button- a (normally0 two position switch in a constant location, the activation of which has a predetermined (but possibly program selectable effect; valuator-- a device which returns a numeric value to the processor; locator-- a valuator pair whose interpretation is that of a coordinate pair; and keyboard-- a large array of pushbuttons, each associated with one or two alphanumeric characters. Two devices, the light pen and the digitizing tablet, will be discussed separately and the rest will be grouped under their functional headings.

The light pen is a photosensitive device, usually in the shape of a stylus with an aperture in the tip, which is capable of interrupting the central processor if it views a light pulse of sufficient magnitude. When it is directed at a portion of the image on a refreshed CRT, the momentary brightening as the beam is swept past the aperture causes the interrupt. The timing of this interrupt with respect to the display cycle allows the processor to determine which part of the image structure was struck. Often this task is facilitated by the display processor returning the coordinates of the strike and the name of the subentity selected.

The light pen obviously implements the pick function. In the case of selection from a set of menu items it is also logically equivalent to push buttons. The valuator function is obtainable through the coordinates returned when touching number lines, and the locator function through the dragging of a tracking cross [4.15] about the screen. This latter construct is necessary if values from dark areas of the screen are required. Finally, as noted above, the light pen can be logically equivalent to a keyboard.

The stylus is another multifunction device, less versatile than the light pen, but better suited to the locator function. A pen-like stylus used in conjunction with a special tablet returns the digitized coordinates of its current location to the processor. Although its obvious application is to allow the user to input drawings, it can also be logically equivalent to pushbuttons if certain sensitive areas of the tablet are defined. A variety of physical principles has been employed in these devices, from capacitive or inductive / magnetostrictive coupling to a grid of wires, through acoustic ranging of a spark pen, to resistive and electrostatic detection. None of these is very cheap ($1500 to $3000 is the typical range).

Interesting variants of the stylus and tablet exist. The Lincoln Wand is a "spark pen in a box" which returns an x y z coordinate triple. The touch sensitive screen, developed at the National Research Council of Canada, uses echo ranging of surface waves in glass to determine the position at which the user touches it. Although the resolution is limited, it is very useful with children.

Returning to functional headings, the pushbuttons usually take the form of regular pushbuttons (dull), touchtone telephone pads, or specialized arrays such as a teleprinter or piano keyboard.

The valuator function is more interesting. Potentiometers and/or digital voltmeters (analog-to-digital converters) are the usual devices. These can be manipulated by knobs and thumbwheels, or even by the human voice, galvanic skin response or electroencephalogram (eg. alpha wave detectors).

Realizations of the locator function are also diverse. The mouse is a handheld object rolled about on any flat surface. The trackball, operating on a similar principle, consists of a ball about the size of a croquet ball which rolls freely in a socket. Thumbwheel pairs are also common. The terminal software or hardware normally displays a cursor or crosshairs to indicate the current position of the locator.

4.5 PROCESSING HARDWARE

The last component of the graphics terminal which will be discussed is the most difficult one to assess. The processing hardware, usually a mini or micro processor but occasionally random logic, coordinates the actions of the other components. Typical duties are: the support of communication protocol (if any; interpretation of display primitives; sifting and formatting of user input; and updating the display file (if any) in response to changes.

In the past two years it has been demonstrated that, as an add-on to make a dumb terminal (such as the Tektronic 4010 series) smart, the small processor more than pays its way, the benefits -- primarily improved response time and reduced communication cost -- are realized in several ways. Interpretation of data base modification commands from the host computer and local display file regeneration [4.10] clearly provides both benefits. Bandwidth compression of geographical data has also been used [4.11] for these purposes. A third example, useful in educational terminals, is syntactic and semantic analysis of text strings in the user conversation as a method of encoding these for reduced communication load.

The obvious candidate for supplying processing power is the microprocessor [4.16]. Its principal virtue is its cost -- typically under $200 for the processor plus peripheral circuitry such as microcode ROM, bus and/or interupt arbitrator, latches, etc., though the bottom line processors sold by minicomputer manufacturers are more expensive. The principal drawback is lack of speed. Although the diversity of architecture makes general speed comparisons difficult, an example of this limitation is interesting. It was observed [4.17] that in a local data base maintenance configuration with a display having internal vector generation hardware, an increase in line speed from 300 baud to 2400 baud made little difference to the speed with which the operator could use the terminal; it was processor limited. Software vector generation (e.g. for the plasma panel) would be even slower, and suitable only for a remote plotter application. Even in this role, though, the reduced communication load due to software interpretation of vector primitives is significant.

REFERENCES

[4.1] James Martin, Teleprocessing Network Organization. Prentice-Hall, 1970.

[4.2] Stephen E. Scrupski, "Distributed Processing Growns as its Hardware and Software Develop", Electronics, vol, 49, no.11, pp.91-97, May 27, 1976.

[4.3] L. Altman, "The 16K RAM is coming", Electronics, vol.48, no.12, pg.80, June 12, 1975.

[4.4] R.E. Matick, "Memory and Storage", in Introduction Computer Architecture, ed. H. Stone, Science Research Associates, 1975.

[4.5] Edward A. Torrero, "Bubbles Rise from the Lab", IEEE Spectrum, vol.13, no.9, pp.28-31, September 1976.

[4.6] "Samples of 92K Bubble Memories Coming from Tl", Electronics, vol.49, no.20, pp.29-30, September 30, 1976.

[4.7] Memory Design Handbook, Intel Corp., pp.9.1-9.30, 1975.

[4.8] L. Altman, "Charge Coupled Devices Move In On Memories and Analog Signal Processing", Electronics, pp.91-101, vol.47, no.16, August 8, 1974.

[4.9] M.A. Copeland, D. Roy, J.D.E. Beynon and F.Y.K. Dea, "An Optical CCD Convolver", IEEE J. Solid State Circ., vol,SC-11, no.1, pp.84-87, February 1976.

[4.10] J. Raymond, "Un language pour terminal graphique intelligent", Proc 4th Man-Computer Communications Conf., Ottawa, May 26-27, 1975.

REFERENCES (Continued)

[4.11] R.D. Kellner and Lynn D. Maas, "A Developmental System for Microcomputer Based Intelligent Graphics Terminals", Computer Graphics (SIGGRAPH-ACM), vol.10, no.2, pp.139-142, Summer 1976.

[4.12] J.W. Brahan et al, "An Experimental Plasma Panel Display Terminal", Proc. 4th Man-Computer Communications Conf., Ottawa, May 26-27, 1975.

[4.13] W.M. Newman and R.F. Sproull, Interactive Computer Graphics, McGraw-Hill, 1973.

[4.14] U. Trambacz, "Towards Device-Independent graphics Systems", Computer Graphics (SIGGRAPH-ACM), vol. 9, no.1, pp.49-52, Spring 1975.

]4.15] R.E. Warburton and H.G. Bown, "Implementation of Tracking Marks for Computer Generated Displays", internal working document, Communication Research Center, Dept. of Communications, Ottawa.

[4.16] Special Issue on Microprocessors, Electronics, vol. 49, no.8, April 15, 1976.

[4.17] J. Raymond, private conversation.

[4.18] Peter Dinh-Tuan Ngo, "A New and Simple Light-Pen Position-Detection Technique for Interactive Plasma-Panel Display Systems", IEEE Trans. Electron, Dey., vol.ED-23, no.9, pp.1058-1063, September 1976.

5. COMMUNICATION REQUIREMENTS

Previous chapters have discussed applications, techniques and hardware of remote graphics, with little consideration given to the nature of the traffic between the nodes. This traffic and its interaction with the standard communication services are the subject of the present chapter. Of particular interest is the suitability of the new public packet switched networks (Datapac, Infoswitch) for various types of remote graphics.

5.1 COMMON CARRIER SERVICES

It is assumed in this report that remote graphics users will not install their own lines, but will subscribe to one of the standard common carrier offerings. There are therefore three general services to be considered: dialup, leased line and packet switching. Each of these should be assessed for graphics work on the basis of error rate, delay and cost. This is emphatically not an easy task because of the complex interactions between distance, data rate, line quality, and connect time in the cost formula (although cost is a non-decreasing function of all of these). Even if one can make sense out of Bell rate schedules, selection of one service depends on application-specific data such as frequency of use, data volume and response time constraints. For these reasons, and because the Datapac rate schedule is not yet firm (as of this writing), no numerical comparisons will be made; instead, simplified models will be used to indicate the dominant parameters in remote graphics work. The following symbols will be used in the models of this section:

R: data rate (bps) of user data terminal equipment
T : connect time of call (seconds)
D: distance between DTEs (host and terminal)
M: message length (bits)
U: fractional utilization of line
C_d: cost parameter of dialup lines ($/sec)
C_1: cost parameter of leased lines ($/month)
r: average rate of generation of messages (no./sec)
T: transport delay in packet network (sec)
C_p: cost parameter of packet switching ($/Kpkt)

5.1.1 Dialup Lines

Dialup lines, both locally and through the DDD network, are the classic form of circuit switching, in which a unique copper path is established between host and terminal.

Cost is fairly simple to establish. Let T be the connect time in seconds and D be the distance of the call. Then:

$$COST = C_d(D)\ T \text{ per call}$$

where the proportionality factory C_d incorporates distance dependence, and equals the asymptotic long call rate. Local calls, on the other hand, cost almost nothing (a small monthly charge).

Delay is also easy. Let R be the data rate in bits/sec and M the average message length in bits. Then the delay from sending of the first bit to receipt of the last bit is (ignoring propagation delay) given by :

$$DELAY = M/R \text{ seconds}$$

Dialup lines are limited to low and medium data rates, which can mean long transmission delays, particulary since M is usually large in graphics work.

Dialup lines are capable of transmitting continuously at rate R, yet during a typical session they are used intermittently. The utilization of the line is given by

$$U = rM/(RT\ )$$

where r is the rate of message generation and M is the average message length.

Error performance is less easy to establish. For low speed (150 bits/sec and less) asynchronous transmission, the Fleming and Hutchinson 1969-70 Connection Survey [5.1] indicates an average character error rate of about 10^{-4}. This figure is of limited interest, of course, because it is difficult to imagine even 300 baud graphics except for highly compressed Scribblephone. Medium speed (1200-4800 bps) synchronous connections, reported in the Balkovic et al 1969-70 Connection Survey [5.2], have an average bit error rate of about 10^{-5}. The major characteristics discovered by both surveys were that the majority of the errors were contributed by a few bad lines (most lines are much better) and that errors were clustered into bursts.

5.1.2 Leased Lines

Leased lines are permanent copper connections between host and terminal. Special conditioning equalizes the spectrum to give the subscriber a range of line qualities to choose from. No dialling is necessary to establish connection, but only one connection is possible.

Cost of leased lines is a flat rate per month, the amount depending on distance and line quality Q:

$$COST = C_1(D,Q) \text{ per month}$$

This figure is independent of usage -- 1 hour per month costs as much as 100 hours per month -- so it is preferable to lease lines rather than dial up if traffic is heavy.

Delay, as for dialup lines, is given by:

$$DELAY = M/R \text{ seconds}$$

again ignoring propagation delay. Because there are no channelizing filters and because of equalization, much higher rates than those of dialup lines are attainable. Voice band rates of 4800 bps and 9600 bps are common, and wideband 50 Kbps is available for very high (concentrated) traffic. The delay is correspondingly reduced.

Error rate is very low (typically 10^{-5} to 10^{-7}) because noisy contacts and switching transients are eliminated and because of equalization.

Leased lines are preferable to dialup lines if usage is over a few hours per day, or if dialup lines cannot meet the delay requirements (because of low speed) or the error rate requirements (which are particularly stringent if data compression is employed).

5.1.3 Packet Switching

Publicly available packet switching will shortly be offered by both TCTS and CNCP as Datapac and Infoswitch, respectively. Although packet switching is well described in books [5.3-5.4] and technical papers, a very brief description will be given here. Users or digital terminal equipment (DTEs) have a dedicated physical connection only as far as a network switch, or node (Figure 5.1). Nodes are interconnected by high speed lines over which user-to-user messages are sent in a time-interleaved fashion, so that these physical lines are shared. Messages originating at a DTE have a format similar to that of Figure 5.2 and contain both addressing and error checking information. The messages are individually error protected by node-to-node detection-retransmission schemes so that received messages are virtually guaranteed to be correct. If they are over a certain length (the maximum packet size), the messages are segmented into packets, which are the basis of the time interleaving, error checking, and storage. The two features -- shared lines and known maximum packet size -- are the source of the economies introduced by this mode of communication.

Packet switching provides an error-free, variable delay virtual circuit between DTEs. Network facilities are required only when the DTE is actually transmitting, so billing can be on the basis of traffic volume, not connect time.

Datapac [5.5-5.6] and Infoswitch [5.7-5.8] provide similar virtual call services, differing substantially only in the matter of call setup. For dumb terminals, the Intelligent Terminal Interface (TCTS) and Infocall (CNCP) will perform the packetizing and call management functions to provide simple users with the cost savings inherent in packet switching. More intelligent DTE's--typically ones handling high volume or a number of simultaneous calls-- implement network end-to-end protocols involving call setup and clearing, flow controls, and the packetizing and sequencing. The TCTS version, standard network access protocol (SNAP) and CNCP's Infogram are similar in this regard.

Cost, as described above, is a function only of traffic volume and the DTE data rate, not of connect time or distance. For Datapac, monthly charges will be on the basis of kilopackets , a normal packet being up to 256 bytes.

$$\text{COST} = C_p \lceil M/2048 \rceil /1000 \quad \text{per message}$$

since M is in bits. Alternatively, an approximate basis for comparision with dialup lines is

$$\text{COST} = rC_p \lceil M/2048 \rceil /1000 \quad \text{per second}$$

Note that the ceiling function $\lceil\ \rceil$ (least integer greater than) means no economies are realized with short packets. This is a significant departure from earlier announcements [5.5] that 32-byte "accounting packets" would be the basic charge unit. To date, the value of C_p has not been announced officially, nor has the very accounting structure of Infoswitch. It is claimed, though, that packet switching will cost less per bit than circuit switching except for high values of dialup line utilization U.

Delay is also more complex in packet switching, being composed of entrance and exit delay associated with inserting and extracting the message from the network, and transport delay within the network. Assuming the rate at which DTEs communicate with the network to be the same at both ends of the virtual link, we have

$$\text{DELAY} = 2MR + T$$

where T is the transport delay. An obvious consequence is that short message delay is dominated by T, whereas long message delay is governed primarily by the DTE data rate. Transport delay objectives for Datapac are quoted as 620 msec (90th percentile) [5.5] and for Infoswitch as 150 msec per node traversed [5.7] (initial Infoswitch configuration is four nodes, fully connected).

Error rate is very low, which for some customers will be a strong attraction. Infoswitch claims a bit error probability of 10^{-12}; Datapac makes no claim, but one can assume that the error rate will be similar.

5.2 GRAPHICS AND THE CARRIERS

This section examines the interaction between remote graphics requirements and the three common carrier services described above. It will be seen that packet switching is, for several reasons, the best choice for most graphics work.

The characteristics of graphic traffic must first be examined. Perhaps the most striking feature is its asymmetry; except for Scribblephone, graphics work involves transmission and display of images generated by the host, with a very light return traffic of user interactions and input.

Also characteristic of graphics is the bulk nature of forward channel traffic. Even Figure 5.3, a comparitively simple frame from a computer animated film, contains almost 500 vectors. Even if they were all "short vectors" (Chapter 6), it would take 1000 bytes to describe the frame. It is not at all uncommon to generate pictures consisting of thousands of vectors.

The third feature of graphic data is its high degree of point-to-point correlation. Smooth curves usually require many closely spaced points for an adequate description-- even short vector format is an extravagant method of transmission.

With these characteristics in mind we can assess the relative merits of the three services. The first conclusion is that asynchronous dialup lines are out of the running because of the data rate limit; the simple picture of Figure5.3 would take between 30 and 60 seconds to transmit at 300 baud, depending on the short vector/long vector mix. At 1200 baud it would still require 7 to 15 seconds. Delay of this magnitude and greater is unacceptable for interactive work. The only situation in which dialup lines would be acceptable is that of bandwidth compressed Scribblephone, since the images evolve slowly.

The comparisons of interest are therefore between synchronous dialup, leased lines and packet switching. If host and terminal are separated by a significant distance (as between different cities) then packet switching is more economical. Within a city, the situation is less clear, and depends on the parameters C_p and C_1 and on the traffic.

Generally, however, packet switching meshes beautifully with graphics requirements in almost every respect. Delay, for example, is not as much of a problem as it is in leased lines, since one can obtain higher entrance and exit speeds relatively cheaply, these being quite independent of the internode trunk speed. There are also several ways in which the charge by data volume of packet switching works to the advantage of graphics. For example, the asymmetry of graphic data streams means that there is, by comparison, negligible cost for the return channel, although the capacity is there if needed. In a similar way, intermittent transmission in the forward channel is reflected in lower costs. The characteristic long messages mean that very few of the uneconomical partially filled packets need be sent. Finally, the very low error rate ensures the safety of bandwidth compressed graphic data, thereby allowing the user to take further advantage of the charge by data volume. One can foresee the cost of transmitting an image becoming dependent on its theoretical information content, rather than on some arbitrary digital representation.

REFERENCES

[5.1] H.C. Fleming and R.M. Hutchinson, Jr., "Low Speed Data Transmission Performance on the Switched Telecommunications Network", B.S.T.J., vol.50, no.4, pp.1385-1405, April 1971.

[5.2] M.D. Balkovic et al, "High Speed Voiceband Data Transmission Performance on the Switched Telecommunications Network", B.S.T.J., vol.50, no.4, pp.1349-1384, April 1971.

[5.3] D.W. Davies and D.L.A. Barber, Communication Networks for Computers, John Wiley, 1973.

[5.4] N. Abramson and F. Kuo, Computer Communication Networks, Prentice-Hall, 1973.

[5.5] W.W. Clipsham and F.E. Glave, "Datapac Network Overview", Proc. Third Int. Conf. Computer Comm., Toronto, August 3-6, 1976.

[5.6] D.A. Twyver and A.M. Rybczynski, "Datapac Subscriber Interfaces", Proc. Third Int. Conf. Computer Comm. Toronto, August 3-6, 1976.

[5.7] Guy F. Carleton, "Infoswitch -- A Public Nationwide Data Network in Canada Provided by CNCP Telecommunications", available from CNCP, February 19, 1976.

[5.8] Guy F. Carleton, "The Infoswitch Network", presented at the Canadian Computer Conference, Toronto, Ontario, October 30, 1975.

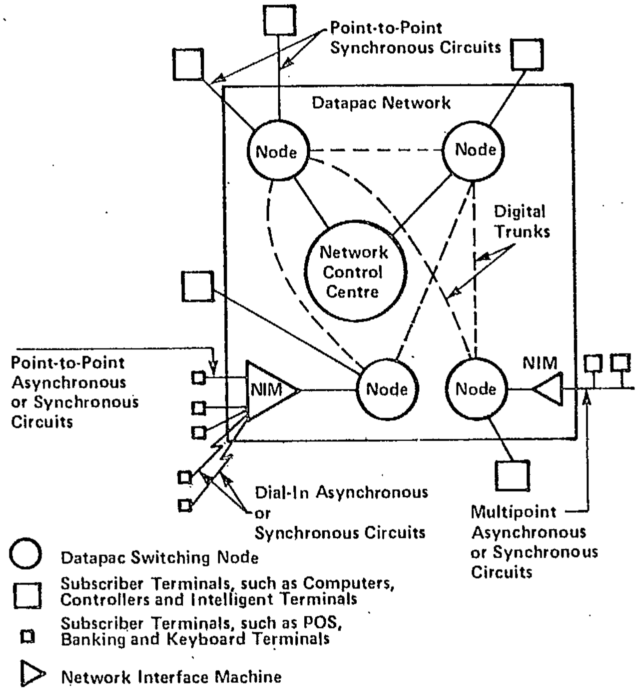


Figure 5.1 Datapac Organization (Ref[5.5])

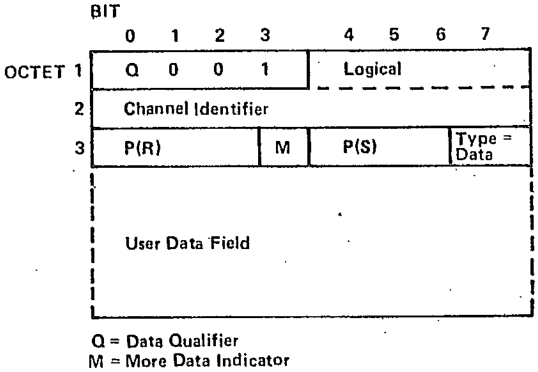


Figure 5.2 Packet Format (Ref(5.5])

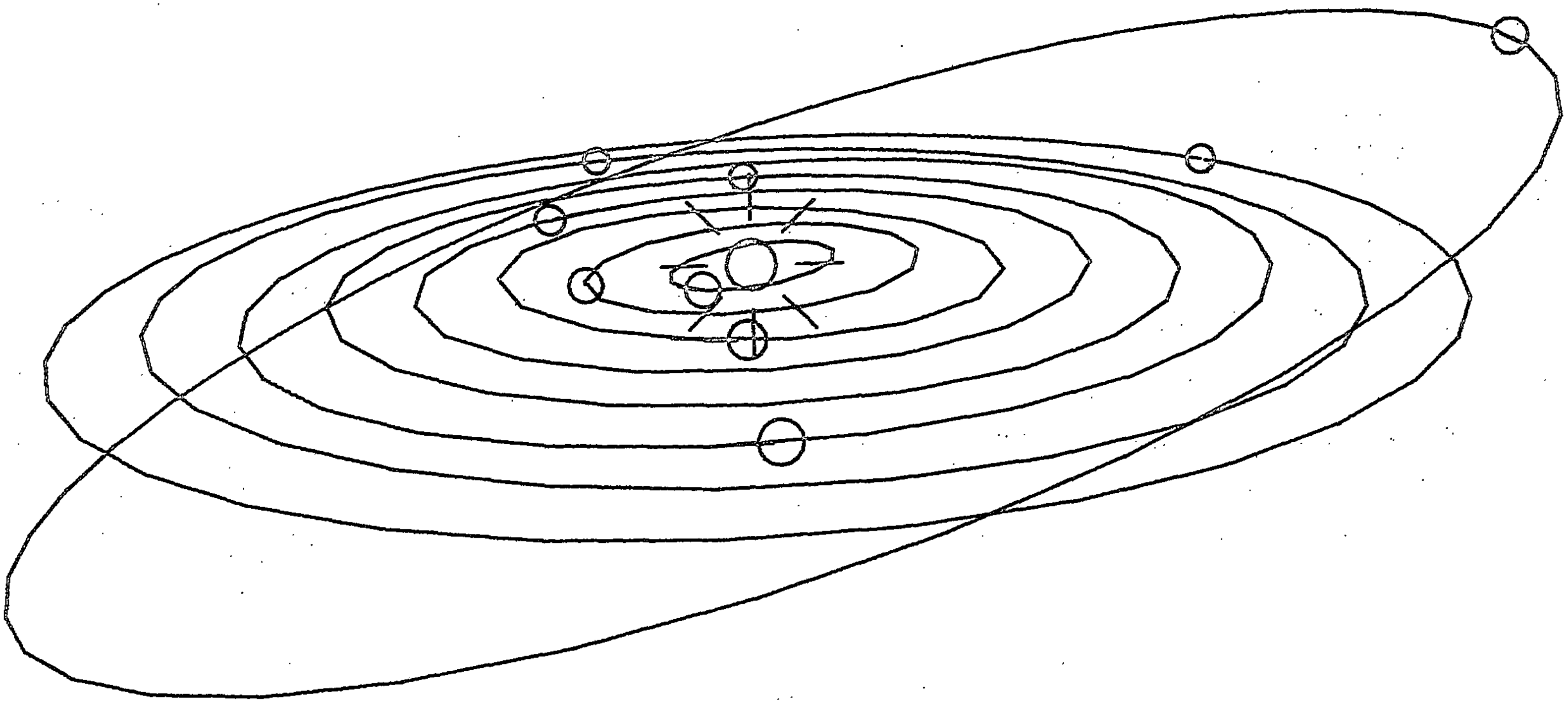


Figure 5-3 Frame From Computer Animated Film

6. TOWARD OPTIMUM CONFIGURATIONS

The last two chapters examined the evolution of graphic terminal hardware, and the effect of the new communication services on graphic transmission. One clear trend stands out from these surveys. The current and projected increase in local processing and storage capability will allow remote graphics systems to take advantage of communication charges based solely on volume of traffic actually transmitted. In this way, delay and cost can be made to approach the information-theoretic limits imposed by the nature and degree of inter-module communication.

Two specific forms of such optimization will be examined in this chapter. The first concerns the allocation of system functions between host and intelligent satellite or, more generally, between the several nodes of a graphic network. The second is data compression, or reduction of redundant data in the transmitted messages to reduce delay and/or required bit rate. Both of these promise fascinating work in the next few years.

6.1 ALLOCATION OF FUNCTION

A large system, such as an application program which accesses a large data base and which requires interactive graphics, is made up of many interacting hardware and software modules. Traditional forms of software organization have recognized only main programs, procedures (subroutines) and interrupt service routines (for the assembly language programmer) as program constructs. More recently, operating systems which support multiprogramming and interprocess communication have weakened these distinctions; any procedure can be restructured as a process, and interrupt service routines can be written to do little more than awaken another process. This view of a system, as a set of (possibly) concurrently executing and communicating processes will be useful to the following discussion, despite any resulting inefficiencies in an actual implementation.

The processes need no longer all reside in the same computer, but can instead be scattered among several nodal sites which have enough intelligence to support multiprogramming.

In a graphics context, the processes involved could be specifically graphical functions such as: display data base maintenance; the transformation, clipping and shading routines associated with creation of a display file; sifting of interrupts from user input devices to result in either direct alteration of the display or formatting of these as asynchronous signals to other processes ; interpretation of virtual display commands; bandwidth compression and so on. Many non-graphical processes may also be involved: communication line protocol; message protocol (as in a remote subroutine call or signal operation); local I/O; main (non-display) data base access, et cetera.

The question now posed is deceptively simple. Given a decomposition of a system into interacting processes, where should each of the processes be placed for minimum communication cost and delay in order to exploit the parallelism latent in multiple processing sites?

6.1.1 Point-to-Point (Two Node) Configuration

We consider first the now classic configuration of a host with a very intelligent graphics satellite. An example of such a satellite is DEC's GT-40 (and others in that series). A fairly simple graphics processor shares memory with and is controlled by a PDP-11 minicomputer. The excess intelligence of the PDP-11 can support many of the functions described earlier in this report, including some portions of the application program itself.

The resulting question of how to partition the modules between nodes has been considered by Stone [6.1] in a general context and Hamlin [6.2] in a graphics context by use of a graph representation (not to be confused with the visual graphics) of the system. In this model (Figure 6.1) each node corresponds to a module, or process, of the overall system, and existence of an arc implies explicit communication between the corresponding pair of processes. The weight given to each arc specifies the cost of separating the two processes by communication lines. Typical interpretations of this weight are communication cost (proportional to the number of words transferred), space overhead taken up by "proxy" processes which handle remote calls, and (less denfensibly) communication and overhead delay. Now if it is assumed that the cost associated with a given partition of the node set into two subsets is given by the sum of the inter-subset arc weights then the minimum cost partition is determined by the minimum weight cutset. This is a straightforward application of any max flow algorithm based on the Ford-Fulkerson theorem --provided that two modules are specified as bound, one to each machine, since the absence of such a constraint obviously leads to a completely centralized configuration. This model and algorithm will be assessed after a description of two operational systems based on it.

CAGES [6.3] is a system developed by Hamlin and Foley at North Carolina, for an IBM 360MS host with PDP11/45 satellite. Briefly, all modules are written in PL/I with little regard to the machine on which they are to run. Binding of modules to machines occurs at compile time, when a preprocessor makes minor alterations in the source code (e.g. Remote procedure calls are replaced by calls to a dummy communication process). Compilation to the target machine code then takes place separately on the two sets of modules.

Experiments were conducted [6.2] on the reconfiguring of some existing application programs through the use of CAGES following min-cut analysis using prior information about communication requirements. Dramatic reductions (up to a factor of 2) were found in response time and communication cost as the satellite was given more functions beyond the benchmark "remote plotter" level. Unsurprisingly, the satellite acquired functions related to display creation and maintenance, while the host was left with large numeric calculations. Because of memory space limitations and lack of a floating point processor on the satellite, however, some configurations were not tried, so the experiments were to some degree inconclusive.

ICOPS, the creation of Stabler[6.4] at Brown University, is similar in intent to CAGES, differing principally in its lack of binding and its non-graphic generality. Since the two processors involved at Brown have the same instruction set, it is possible for them to migrate the modules at run time. A "meta-level" monitors inter-module traffic and periodically recomputes the optimum partition and shifts modules when appropriate. ICOPS was applied to graphics by Michel and van Dam[6.5]. They found that the dynamic assignment inherent to this system produced the movement of clusters of modules at a time as the host processor availability changed. Also interesting was the fact that the first modules to move to the satellite were not those associated with display code generation. This was apparently due to a peculiarity in their data structures, but it serves as a warning against hard rules.

Both CAGES and ICOPS are based on the graph model of separation costs. However, the optimization algorithm and the model itself have deficiencies. Regarding use of max flow algorithms, memory size constraints can not be included except in an iterative or backtrack modification. It is also unsuitable for extension to three or more processing sites-- a min-cut ternary or n-ary partition algorithm is still to be devised. With respect to the model, one deficiency is that it emphasizes the cost of separation without considering the possible gain due to parallel processing. Another problem is that delay is not really suitable as an arc weight, since the delay incurred by partition equals the sum of the arc delays only if the transmissions are strictly sequential. In a good multiprogramming environment, however, asynchronous operation and overlapped transmissions destroy this synchronism. One concludes that this method of allocating functions still has a long way to go.

6.1.2 The Network Configuration

In this section we briefly consider a network which has nodes of roughly equal intelligence and an active human user at each. The problem is to implement a "shared graphic space" of the type being explored by Bown and Sawchuck [6.6]. The shared graphic space is in fact a shared data base with updates from several sources.

Should the data base be partially or completely segmented between the nodes, or should it be replicated at every node? If the former, then display file update commands must be broadcast following a data base update at a single node. If the latter, then the data base update commands must be broadcast. This latter, replication of the data base, appears preferable for two reasons. First, the data base update commands are normally more compact than the display file update commands, which can involve replacing large sections of the image. Second, data base updates have a less rigidly defined format than display file updates, so that bandwidth compression can be applied. The operating assumption is, of course, that it is preferable (and possible) to use more memory space than to suffer more communication cost and delay.

Distinct from the problem of carving up the data base is that of regulating access to it. Certainly completely asynchronous updates must be prohibited, at least on the same entity, where the interleaving of updates would produce garbage. An extreme case is that of the inking, or scribblephone function; all other users must be locked out of that portion of the data base, at least, during the relatively long time a line is being drawn, in order to prevent interleaving of vectors. These problems are really those of controlling access to any distributed data base, graphic or not.

Controlling access to a distributed data base is receiving increasing attention [6.7-6.9]. A well known method of protecting shared data structures on a uniprocessor system or multiprocessor shared-memory system is the use of semaphores [6.16]. This mechanism, by which processes can suspend themselves while waiting for a resource, is complicated in the multinode system by communication-induced race conditions. Two subproblems can be distinguished. The first is that of implementing software locks. The second is that of preventing deadlock, a condition in which, because of the sequence of requesting non-preemptable resources, two or more processes are unable to proceed even though the total resources of the system would permit them to continue alone, one after the other. Both of these problems can be approached as a centralized or as a distributed control (the feasibility of the latter was demonstrated only recently [6.9]). The primary differences between the two resulting structures are that distributed control is less vulnerable to total failure, but it requires the control to be replicated at every node and it requires significantly greater overhead in the form of number of control messages and synchronization delay. For the shared graphic space scenario, it is therefore clear that centralized control is to be preferred.

A further observation on the application of general results to the graphic network described above is that deadlock, in the sense of a circular wait induced by processes sequentially requesting control over network files, is impossible in the centralized control. This follows since there is only a single multicopy data base to be accessed.

Both central and distributed control would benefit if the communication network provided a multi-address, or "broadcast", facility, since so many duplicate messages are sent. Without this facility, the graphic terminals must assume the responsibility of sending sequential messages, each of which incurs the network entrance delay. Neither Datapac nor Infoswitch have incorporated the broadcast feature.

In summary, an interactive graphic network should replicate the graphic data base at every node. Update access to this data base should be regulated by a centralized control structure. The control will probably be simple enough to be activated at another node in case of control node failure.

6.2 DATA COMPRESSION

The introduction to this chapter indicated that the increasing availability of cheap intelligence (processing power plus memory) would allow more complex terminals to reduce the communication cost and delay. As an example, the allocation of system functions was considered in Section 6.1. This section discusses another optimization, data compression.

Data compression is sometimes called bandwidth compression or redundancy reduction. The objective is to transform one description of a data structure into another, in such a way that the number of symbols is reduced, but the original description can be regenerated from the transformed one. In other words, the number of binary digits is reduced, but the number of information-theoretic bits remains the same, thereby easing storage requirements. In the communication context, this reduction translates to either lowered data rate (bandwidth) requirements for the same transmission time, or reduced transmission delay at the original data rate (or similar variations). In both cases, a direct communication cost saving is realized, since the packet switched networks charge roughly on the basis of number of bits transmitted (Chapter 5).

Non-graphic information can be subjected to data compression, but since it is of less interest in this report, it will be dealt with summarily. Examples of such compression are the storage of predefined formats for form-filling applications, storage of numbered canned responses such as error messages or the 1000 most common words, and the results of syntactic analysis/synthesis of restricted natural language in CAI applications.

Specifically graphic information has possibilities for compression which are distinct from those of text. To explore these, we must again make the distinction between line drawings and filled-area images. A similar distinction was made in Chapter 4 regarding vector and raster displays; these are in fact normally associated with line drawings and filled-area images, respectively. It is important to note, though, that at this point the discussion is concerned with the images and their descriptions; these have little to do with the actual display technology used.

Since several more distinctions among image types and descriptions will be made, Figure 6.2 is presented to the reader as an aid to keeping them straight.

One final point -- the well-known fragility of compressed images is not a problem if the commercial data networks (Datapac, Infoswitch) are to be used, since these provide a virtually error-free connection.

6.2.1 Line Drawings

Line drawings consist of a sequence of plane curves, each with qualifying information such as colour, intensity, line texture, etc. Although such curves can be described in many ways mathematically, the most convenient is usually parametrically, as a pair of functions x(t) and y(t) where the parametric variable t is usually in the range $0 \leqslant t \leqslant 1$.

There are two general ways to define the functions x(t) and y(t). The first is algorithmically, as a combination of standard mathematical functions or, more precisely, as a sequence of operations to be executed in order to compute x and y from t. The second way is empirically, as a sequence of parameters to be used in a function of predefined or implied form. As an example, consider the definition of a unit circle. An algorithmic definition could consist of the functions (which are usually embedded in a display procedure) $x(t)= \cos 2\pi t$ and $y(t)= \sin 2\pi t$. An empirical definition could be a sequence of x,y pairs on the circle to which some pre-specified form of interpolation is aplied. Algorithmic descriptions are usually a form of data compression themselves, and direct transmission of these descriptions appears less impractical when one considers the present commercial availability of portable APL interpreters. Normally, though, one is confronted with empirical definitions, and it is with these that the discussion on data compression is concerned.

We shall take as the benchmark empirical definition a sequence of absolute coordinates along the curve:

$$\underline{u}_i , \ 1 \leqslant i \leqslant N$$

where the vector $\underline{u}_i = (x_i , y_i)$ and N is the number of points in the definition. That this is usually a highly correlated sequence has been recognized for years. Most display equipment will interpret the first backward difference sequence ("relative vectors"):

$$\underline{v}_i = \underline{u}_i - \underline{u}_{i-1}$$

Since these vectors are usually short, a special storage and transmission format, "short relative vector", is employed if each component of the relative vector fits into a half word. This in itself realizes a compression factor of 2.

Variations on the short relative vector are common. The Tektronix 4010 series attempts further compression with relative vector formats of 1,2,3 or 4 bytes depending on the size of each component. The improvement relative to the basic 2 byte short vector format is questionable. The BNR prototype Scribblephone divides the screen into 12 zones with special format relative vectors used as long as the absolute location lies within a single zone (a rather bizarre scheme, reminiscent of the old PDP-8 memory pages).

As an alternative to short vectors, one can fit curve segments from a predetermined family and store or transmit the defining parameters. The Hughes Conographics terminals, for example, contain display hardware to generate conic section curves, and are advertised as providing data compression in this way. An often-quoted example in their literature is a sleek sports car defined with only 205 words. Hughes claims that an equivalent rendition using straight line segments would require 4800 words--an impressive comparison, though one wonders about the accuracy criterion imposed on the straight line version.

We saw in these examples two distinct methods of exploiting the redundancy of the $\underline{u}_i$ sequence. The first is predictive encoding, in which each point is encoded with a reduced number of bits which describe its location relative to that predicted on the basis of preceding points. In the case of short relative vectors, the prediction is that each point is the same as the preceding one. The other method is that of spline fitting, where a spline is a plane curve with parameters chosen to fit the data. Conic, polynomial (including linear), and Fourier splines are the most common. The main difference between the two methods is that predictive encoding forms a value for each point, employing a possibly variable number of bits; spline fitting, on the other hand, uses a fixed number of bits for the parameters of a curve which represents a variable number of data points (the exact number depends on how closely the data resembles members of the curve family). Of the two methods, spline fitting appears to offer greater possibility for data compression because of its implicit use of the correlation of a point with both its neighbours.

6.2.2 Scribblephone

Data compression for scribblephone has been investigated by the author and his students. This application has conflicting requirements of low bandwidth and low delay, but, as suggested in Chapter 2, it may become a significant extension to conventional telephony.

Scribblephone offers opportunities for further analysis because the sequence of points has well developed statistics, unlike those of more general pictures. In fact if the x and y processes are considered to be independent (experimental evidence [6,10] shows this to be an excellent approximation) then one can directly apply many of the classic one dimensional predictive encoding techniques (e.g, Differential PCM). There is one significant difference-- the source is inherently amplitude discrete (digitization normally occuring in the stylus location hardware) and can therefore be compressed and reconstructed with zero error.

The general form of the predictive encoder is shown in Figure 6.3. The variable u_i is shown as a scalar to represent either x_i or y_i, but it should be remembered that both x_i and y_i must be encoded. If the predictions $\hat{u}_i$ are reasonably accurate, then the error sequence e_i has much lower variance than the u_i sequence and requires fewer bits to specify its value. Formatting the error, which is an integer number of picture elements, for transmission can consist of byte, nibble (4 bit) or bit stream oriented mapping as well as insertion of new line, clear screen and other signals. Reconstruction at the receiver is accomplished using the same predictor as at the transmitter.

For the remainder of this discussion the predictor will be restricted to be linear, to take advantage of the rich theory of stochastic processes and linear dynamic systems. The prediction is the familiar discrete convolution plus an additive constant to absorb non-zero mean values:

where the maximum order of the predictor is 3 simply because experiment has shown this to be more than sufficient. Rather than continue with this conventional form for predictive encoding, we observe that $e_i = u_i - \hat{u}_i$ is the difference of two close numbers, so that relative errors in the calculation of $\hat{u}_i$ will be magnified. Further, with 9 or 10 bit values of u_i the required accuracy in the calculation of the terms of the predictor will not be available on 8 and 16 bit processors without resort to awkward multiword integers or, even worse, conversion to multiword floating point. We therefore switch to a mathematically equivalent, but numerically superior expression. Define the first and second backward differences v_i and w_i of the u_i sequence:

$$\begin{aligned} v_i &= u_i - u_{i-1} \\ w_i &= v_i - v_{i-1} \\ &= u_i - 2u_{i-1} + u_{i-2} \end{aligned}$$

and the pseudo state vector $\underline{s}_i$:

$$\underline{s}_i = (1, u_i, v_i, w_i)'$$

where the constant 1 extends the state vector to make later expressions more compact. The prediction is performed in two stages:

$$\hat{v}_i = \underline{a}'\underline{s}_i$$

and

$$\hat{u}_i = u_{i-1} + \hat{v}_i$$

where $\underline{a} = (a_0, a_1, a_2, a_3)'$. Why is an improvement observed? First, a small difference $\hat{v}_i$ is estimated directly, and second, the components v_i and w_i of $\underline{s}_i$ which are also small differences, can be computed exactly since the source is amplitude discrete (although linear prediction ignores this property).

The prediction coefficients, the components of a, are selected to minimize the mean square prediction error $e_i = (v_i - \hat{v}_i)$. The usual formulation of such regression problems leads to the following requirement on a:

$$R\,\underline{a} = \underline{w}$$

where the autocorrelation matrix $R = \overline{\underline{s}_i \underline{s}'_i}$, the vector $\underline{w} = \overline{v_{i+1} \underline{s}_i}$ and the index i is unimportant since we are treating a stationary process. As a point of interest, the values of R, $\underline{w}$ and $\underline{a}$ computed from a recent experiment [6,10] are shown in Figure 6.4. The coefficient a_3 is small; the granularity of the writing space proves to be too much for useful third order compression.

A block diagram of the operations involved in the predictive encoding process is shown in Figure 6.5. Although the structure appears complex, the software required to realize it is not particularly large or complex. The sequence of operations is simply:

1. Update: calculate $\underline{s}_i$ from $\underline{s}_{i-1}$, u_i
2. Predict: $\hat{v}_{i+1} = \underline{a}'\underline{s}_i$
3. Increment i; go to 1

Now for the compression. The probability density function of the error e directly determines the minimum number of bits required for transmission. If e is rounded to the nearest picture element then a simple entropy calculation on some experimental data [6,10] yields 4 bits/sample. This figure assumes 60 Hz sampling of pen position on an 11"x11" surface with linear resolution 100 points/inch. There remains the problem of approaching the entropy limit with a practical formatting scheme. A simple system has realized 8 bits/sample (480 bit/second) experimentally.

Is real time transmission of handdrawn graphics possible at teletype rates over ordinary lines? One possibility is to buffer transmission to take advantage of intervals in which the pen is up. Lowered spatial and temporal resolution is another easy way, but better methods should be devised.

6.2.3 Filled Area Images

Filled area images, similar to those normally seen on film or television, consist of patches of constant or smoothly varying intensity and/or hue. These make an image of, perhaps, a face, a car or a building, with a realism impossible in the more primitive line drawing [6.11-6.12]. Inexpensive raster-type terminals capable of displaying these images are becoming more common (Chapter 4), but the display itself is the least of the problems in this type of remote graphics.

How can these images be described? This is not a simple question, since it involves rendering a two or three dimensional scene as a one dimensional string of symbols--even novelists can have trouble with this. Two distinct possibilities suggest themselves: a structured description of the image as a list of the basic elements and the spatial relations among them; alternatively, a structureless description as a list of the intensity/hue at regularly spaced points (picture elements or pels) arranged in raster fashion. Clearly, transmitting the latter requires less intelligence of

Finally, we consider transmission of the unstructured description in the case of smooth variations of intensity/hue within elements of the scene, such as that produced by shading of three dimentsional objects. Gouraud shading [6.17], the simplest and fastest area filling algorithm, results in a linear variation of intensity with distance. Compression can be acheived with a variation on run length encoding (RLE) : each segment of the raster line is described by the initial intensity/hue, the final intensity/hue and the length. An appropriate name for this encoding is first order RLE (FRLE) to distinguish it from zero order RLE (ZRLE) described above. There is an obvious relation to zero order and first order hold systems used in picture bandwidth compression [6.15].

Reconstruction of FRLE images at the terminal is not as simple as for ZRLE. The intensity dot on the raster line segment must be calculated and placed in the display file as a single dot.

A significant weakness in today's graphic hardware is exposed by these comments. To the author's knowledge, no manufacturer provides hardware linear interpolation of intensity. This deficiency could be remedied by treating endpoint descriptions of vectors as triples: (x,y intensity). Two line drawing primitives are suggested:

```
DRAWABS(X1,Y1,I1,X2,Y2,I2)
DRAWREL(X,Y,I)
```

where linear interpolation is performed on all three components.

No estimates of RLE efficiency have been given, as no data currently exists on the statistics of computer generated filled-area images. This data must be obtained before standardization of a block size for RLE.

REFERENCES

[6.1] H.E.Stone, "Multiprocessor Scheduling with the Aid of Network Flow Algorithms,"submitted IEEE Trans Software Eng. 1976.

[6.2] G.Hamlin jr., "Configurable Applications for Satellite Graphics," Computer Graphics (SIGGRAPH-ACM), vol.10, no.2, pp.196-203, Summer 1976.

[6.3] G.Hamlin and J.D.Foley, "Configurable Applications for Graphics Employing satellites," Computer Graphics (SIGGRAPH-ACM), vol.9, no.1, pp.9-19, Spring 1975.

[6.4] G.M.Stabler et al, "An Approach to Distributed Computing," submitted to Comm. ACM, 1976.

[6.5] J.Michel and A.van Dam, "Experience with Distributed Processing on a Host/Satellite Graphics System," Computer Graphics (SIGGRAPH-ACM), vol.10, no.2, pp.190-195, Summer 1976.

[6.6] H.G.Bown and W.Sawchuck, "An Interactive Visual Communication System," submitted to Computers and Graphics, 1976.

[6.7] P.F.King and A.J.Collmeyer, "Data Base Sharing -- An Efficient Mechanism for Supporting Concurrent Processes," Proc. AFIPS Nat. Comp. Conf., pp.271-275, 1973.

[6.8] G.M.Booth, "The Use of Distributed Data Bases in Information Networks," Proc. First Int. Conf. On Computer Communication, pp.371-376, Washington, D.C., October 1972.

[6.9] S.A.Mahmoud and J.S.Riordon, "Protocol Considerations for

Software Controlled Access Methods in Distributed Data Bases," Proc. Int. Symp. Computer Performance Modelling, Measurement and Evaluation, (ACM-IFIP), pp.241-264, March 1976.

[6.10] M.A.Dagenais, "Bandwidth Compression for Digital Transmission of Hand-Drawn Graphics," M.Eng. Thesis, Carleton University, Ottawa, 1976.

[6.11] R.D.Resch, "A Portfolio of Shaded Computer Images," Proc. IEEE, vol.62, no.4, pp.496-502, April 1974.

[6.12] Donald P. Greenberg, "Computer Graphics in Architecture," Scientific American, May 1974.

[6.13] J.Capon, "A Probabilistic Model for Run-Length Encoding of Pictures," IRE Trans. Inf. Th., pp.157-163, December 1959.

[6.14] S.Golomb, "Run Length Encodings," IEEE Trans. Inf. Th., vol.IT12, pp.399-401, July 1966.

[6.15] C.M.Kortman, "Redundancy Reduction -- A Practical Method of Data Compression," Proc. IEEE, vol.55, no.3, March 1967.

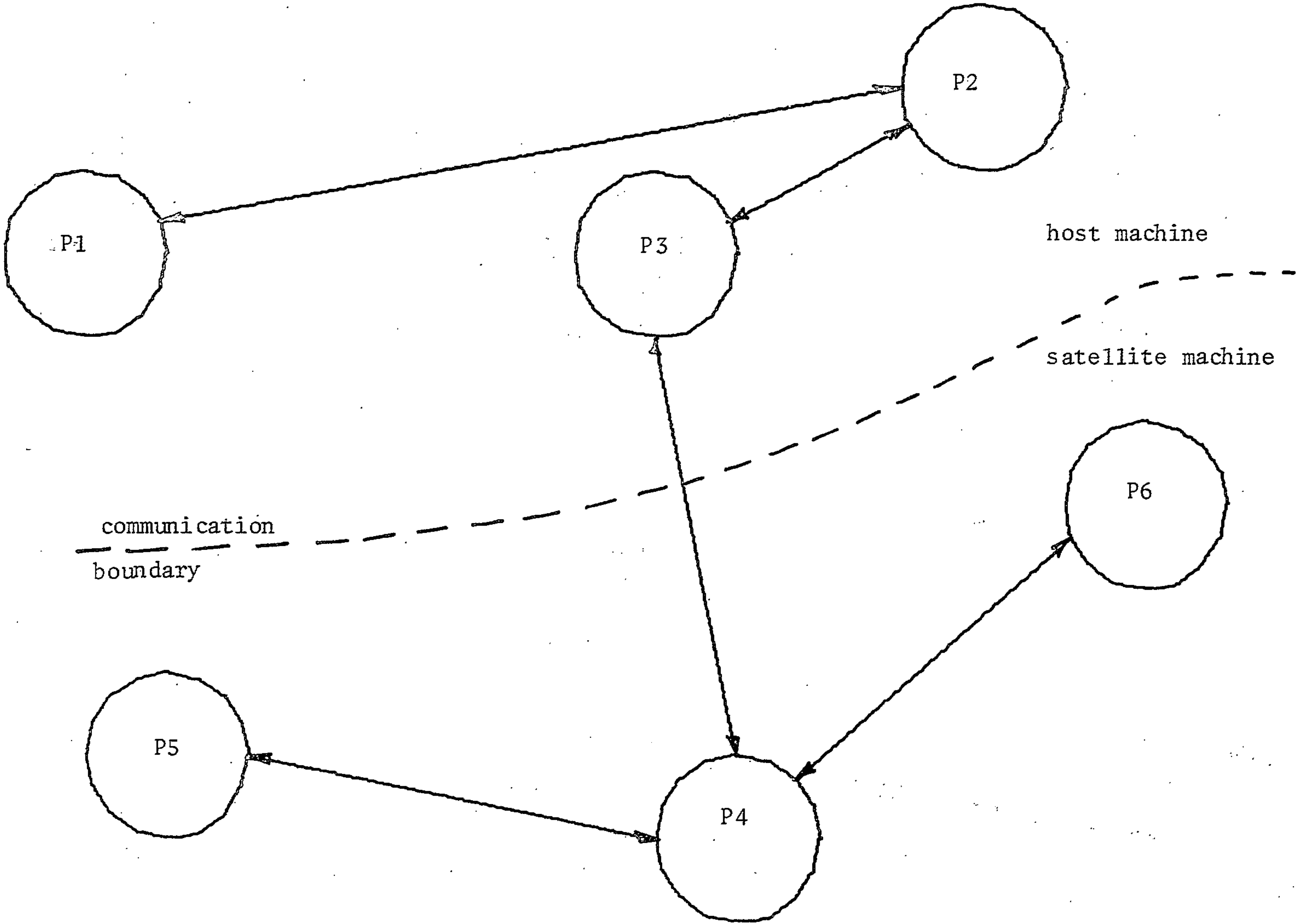


Figure 6.1 Graph Representation of Dichotomized Processing

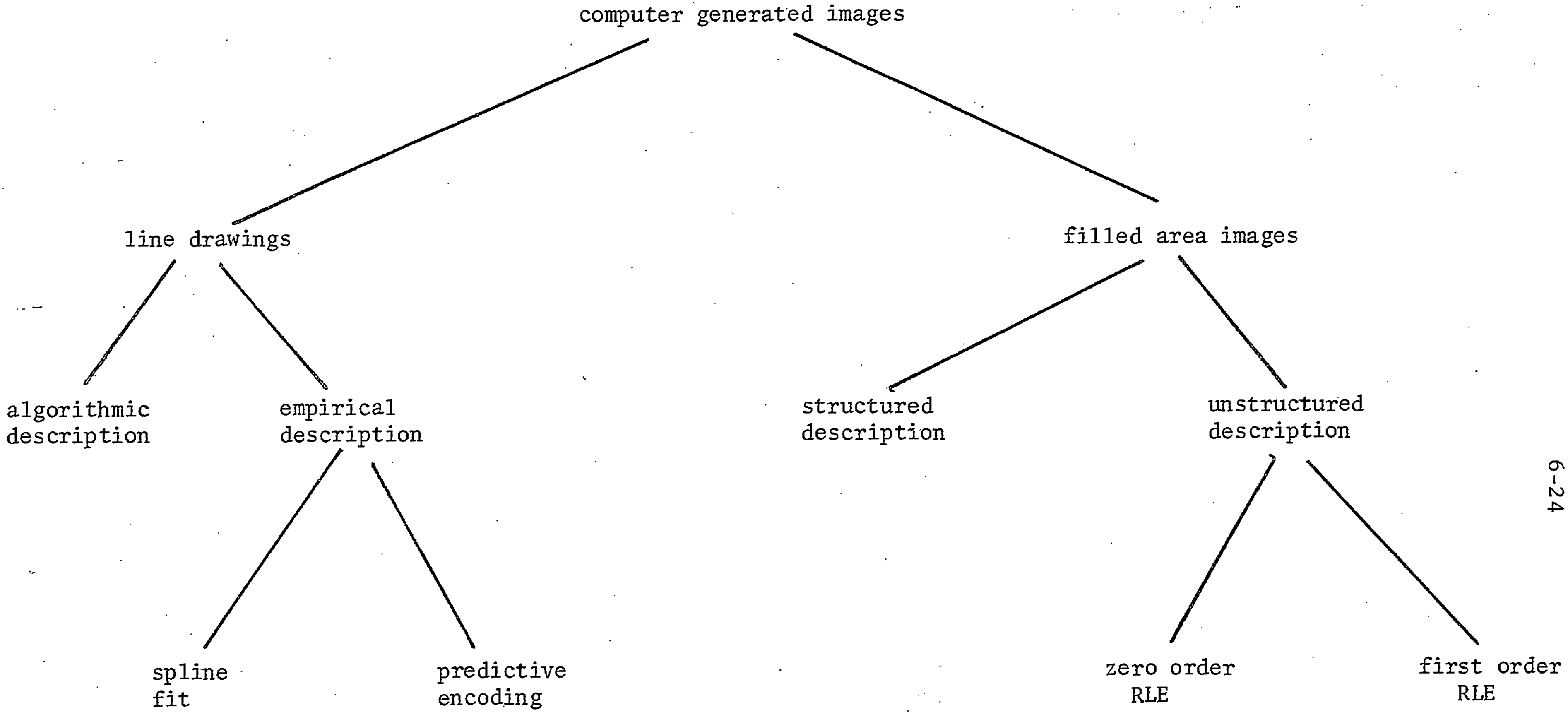


Figure 6.2 Hierarchy of Image Descriptions

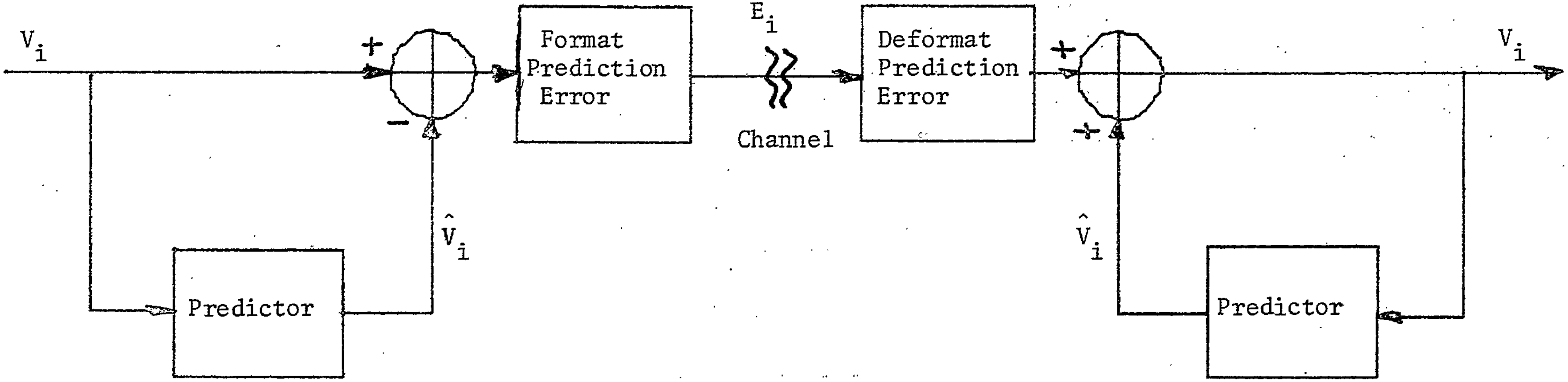


Figure 6.3 General Structure of Predictive Encoder

$$R = \begin{bmatrix} 1 & -30 & .59 & 0 \\ -30 & 33044 & -27 & -9 \\ .59 & -27 & 15 & -3 \\ 0 & -9 & -3 & 6 \end{bmatrix} \quad \underline{a} = \begin{bmatrix} .019 \\ 6.5\text{x}10^{-4} \\ 1.0 \\ 1.0 \end{bmatrix} \quad \underline{w} = \begin{bmatrix} .59 \\ -15 \\ 12 \\ 3 \end{bmatrix}$$

Figure 6.4 Experimental Parameters Describing Hand-Drawn Graphics

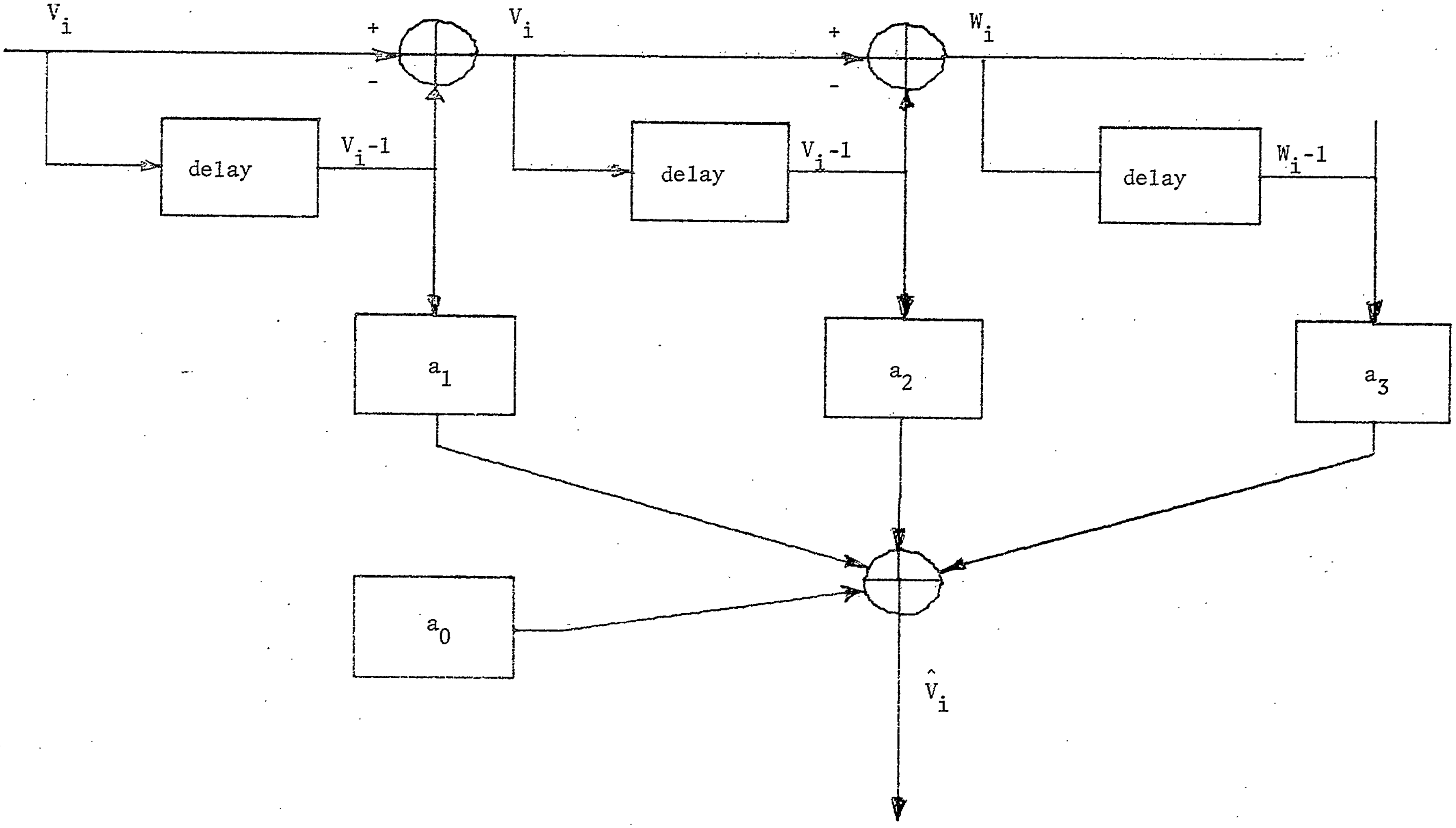


Figure 6.5 Structure of Scribblephone Encoder

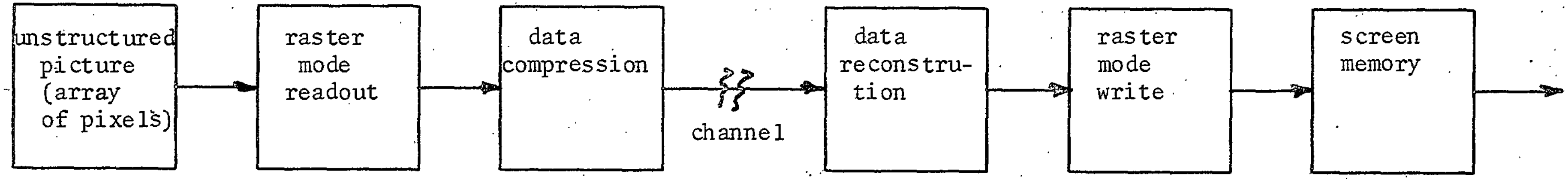


Figure 6.6 Transmission of Unstructured Picture Description

7. SUMMARY

Remote computer graphics is in the middle of a revolution. Technological changes in communication systems and in electronic devices have combined to make possible a graphics terminal which is low in capital and operational cost and high in quality.

Computer graphics will become more and more a part of our lives as these developments seep into the marketplace. Six roughly defined application areas were discussed in Chapter 2: education, scribblephone, process monitoring, technical and managerial information display, real time graphic networks and entertainment. Although the first five can not be expected to represent a significant portion of network traffic, they are nevertheless vital areas in which information technology will proliferate through all levels of society. By analogy, one can note that data does not now represent a significant portion of the circuit-switched network traffic, and that expenditure on community programming is a totally negligible fraction of the cost of programming presented on CATV networks. Yet no one would deny the importance of data traffic or of community programming.

The sixth area, entertainment, presents some rather unsettling questions. The natural outgrowth of the video game is the cheap graphics terminal hooked to a timeshared computer through the telephone network. New services will enter the home through the telephone, thereby making some of the CRTC deliberations on the role of the cable companies slightly irrelevant. More important is the distortion in traffic patterns caused by the typical long call associated with data. Is the telephone network capable of handling a major increase in the number of such calls?

The rapid evolution of semiconductor technology is providing opportunities for efficiency impossible before. As the existing technologies become cheaper, new ones are being introduced almost yearly. The principal gains are in memory speed and capacity, and in microprocessing power. In the graphics area, we see cheaper and/or more complex terminals being constructed. More complex images can be displayed, and the terminal can incorporate higher levels of picture structure. This has two main effects: the host is offloaded of many chores (a form of parallel processing) and the communication lines are better utilized.

As an example, workers at Stanford recently announced their design of a low-cost video graphics terminal [7.1]. Intended as a replacement for the Tektronix 4010 remote plotter type of storage tube terminal, it provides interpretation of point, vector, and character primitives. Semiconductor memory provides storage for the output image which is video compatible, with 640x480 resolution. It can also be used as a conventional alphanumeric terminal, in which case an internal video character generator is driven by the memory, now acting as a text buffer capable of holding 600 80-character lines (about 50% more than Chapter 6 of this report). The 19 lb.

device cost $2000 to assemble (1975 prices) and is expected to drop to $1000, which invites comparison with the $6000 and up price of the 4010.

Although the extended function video terminal will clearly be the dominant force in the remote graphics market for the next few years, its resolution is limited. For high resolution (1024x1024) applications the quadrupling of screen memory requirements and the cost of non-standard 1029 line video equipment may make it uncompetitive with the plasma panel based terminal, especially now that light pen interaction is possible on the panel. However, only video terminals can generate colour cheaply, and this may well prove the deciding factor.

The parallel trend in communication technology is the introduction of public packet switching by TCTS (Datapac) and CNCP (Infoswitch), promising significant savings for most forms of data traffic. For graphics, the key factor is the charge by data volume, instead of by connect time. This provides further motivation for the exploitation of cheap intelligence at the terminal to implement communication protocols and to perform data compression.

The use of newly available terminal intelligence was examined in Chapter 6. Methods of partitioning functions between host and terminal were seen to be potentially profitible, but are still in a relatively undeveloped state. A look at forms of data compression relevant to this study turned up two interesting facts. First, scribblephone should be possible at low rates (300 baud). Second, there is a need for a new graphic primitive to reduce communication load: the simultaneous linear interpolation of x, y, and intensity in the drawing of vectors. This is not yet implemented on any terminal, to the author's knowledge.

REFERENCES

[7.1] F.Baskett and L.Shustek, "The Design of a Low Cost Video Graphics Terminal," Computer Graphics (SIGGRAPH-ACM), vol.10, no.2, Summer 1976.