UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

Computer Graphics Laboratory Fall 1983 Review

Richard H.Bartels
John C. Beatty
Kellogg S. Booth
Daniel E. Field

CS-83-33

December, 1983

Computer Graphics Laboratory Fall 1983 Review

Richard H. Bartels
John C. Beatty
Kellogg S. Booth
Daniel E. Field

Computer Graphics Laboratory
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
(519) 886-1351

Technical Report CS-83-33

ABSTRACT

The Computer Graphics Laboratory at the University of Waterloo is a group of students and faculty doing research within the Department of Computer Science, and affiliated with the Institute for Computer Research. Interests include most areas of computer graphics, with particular emphasis on the modeling of freeform curves and surfaces, scan conversion and anti-aliasing algorithms, visible surface computation, illumination models, the psychophysics of vision, the psychology of man-machine interfacing, and interaction methodologies.

We describe the computer science courses directly related to computer graphics, summarize the hardware available in the Computer Graphics Laboratory, and survey recent and current research by members of the group. Future research and hardware acquisition plans are reviewed as well.

The research reported here was supported in part by the Natural Sciences and Engineering Research Council of Canada, the Province of Ontario's BILD program, the National Science Foundation, the University of Waterloo and the Computer Graphics Laboratory.

Contents

Introduction

The University of Waterloo Computer Graphics Laboratory, established in 1979, consists of four faculty members (Richard Bartels, John Beatty, Kellogg Booth and Dan Field) and about twenty students who share a VAX 11/780 computer, Adage/Ikonas RDS-3000 raster display, Evans & Sutherland MPS vector display and related equipment. Support services are provided by a full time laboratory manager (David Martindale) and an administrative assistant (Adele Newton).

Members of the Laboratory regularly teach an introductory course and two advanced seminars in computer graphics. Students working in the Lab are expected to have taken the former, and generally take at least one of the latter.

Research in the Lab is primarily centred on the modeling of sculpted surfaces, interaction techniques, hardware for scan conversion and anti-aliasing, visible surface computation, illumination models and programming methodology. Most of the students involved are pursuing a masters or a doctorate in computer graphics in conjunction with faculty research.

In order to avoid disrupting work in the Lab, and to ensure that appropriate personnel can be present, tours and demonstrations are conducted only on Fridays and must be scheduled at least a week in advance through Adele Newton.

Subsequent sections of this report describe the graphics classes we teach, the hardware environment, research activity, mechanisms for joining the Lab, student support policy, and future plans for the Laboratory. This document updates information provided in [Beatty81a].

Teaching

Introductory Computer Graphics – CS 488/688

The aim of this course is to supply a "complete" overview of computer graphics appropriate for people who wish to understand how graphics software is actually implemented and interfaced to display equipment. We cover basic hardware, software and human factors, discuss current trends in computer graphics, and briefly survey some of the flashier research currently being done in visible surface processing and anti-aliasing. Students read a number of published papers as well as the text (*Fundamentals of Interactive Computer Graphics* by Foley and van Dam).

We believe in "learning by doing," so each student implements a graphics package which supports a linked list data structure for describing images, perspective and orthographic projections from an arbitrary eyepoint, scaling, rotation, translation, windowing, three forms of text, clipping, viewporting, and a simple z-buffer visible surface algorithm. The package is modular and device independent. Working on a VAX 11/780 running Berkeley Unix (the Math Faculty's "watmath" machine), students write about 4,500 lines of Pascal and implement four device drivers. The work is spread over nine assignments, handed out at the beginning of the term so that students may plan ahead. In the last assignment students are asked to do something – anything! – with their package (see Figure 1).

Although we are teaching graphics, not software engineering, students taking the course often report that the chance to write a software system of this size is a valuable experience. The Pascal teaching package is similar to the C graphics package used for our research, and has served as the starting point for a number of student research projects. Graduate students who want to specialize in graphics often find it desirable to serve as a teaching assistant for the introductory graphics class to increase their familiarity with the package.

CS 488/688 is evolving in several ways. Firstly, the University has supplied us with graphics tablets for use in the class, and a graduate student (Klaus Ahlers) is designing a tablet device driver and menu package for use by CS 488/688 students. We are also evaluating the suitability of Keynote Design's KD404 terminal (a Volker-Craig look-alike with 240x256 by 4-bit-plane graphics) for use as the primary output device in CS 488/688. Finally, we are moving towards greater compatibility with the GSPC Core approach to viewing three dimensional scenes.

CS 488/688 is offered every term, and is available both to undergraduates and graduates. Class sizes vary between 30 and 60. It is one of the more time-consuming fourth year courses offered by the Department of Computer Science. The assignments are generally done in groups of two, although in terms with light enrollment students may be asked to do them individually. The organization of CS 488/688 is discussed in [Beatty81b].

Topics in Computer Graphics – CS 788

This course provides an in-depth examination of various topics beyond the scope of the introductory computer graphic class. Many of the lectures are given by visitors, who discuss their particular research interests. Hence the course content varies from year to year, depending on the current interests of the instructor or instructors involved. Past topics have included geometric modeling using non-polygonal patches (such as Bezier, B-spline and Beta-spline techniques), high performance hardware (especially parallel hardware for raster display and VLSI chips for doing graphics), anti-aliasing algorithms, interaction techniques, the use of colour, and computer animation.

CS 788 does not generally involve programming. Instead, students read extensively from the literature and are expected to analyze research problems – often with the goal of finding a masters or doctoral thesis topic. A term essay is required which involves analyzing a problem, examining the literature for related previous work, and proposing a line of attack on the problem. The term essay often serves as a first draft for the opening chapter(s) of a subsequent masters essay or thesis.

The introductory graphics class is usually a prerequisite for the topics class, partly because we assume the material covered there as background, and partly because it is often convenient to cast the 788 term essay as an addition to the basic graphics package.

CS 788 has traditionally been taught in the Fall term of each year, although in the future it will probably be moved to the Winter term.

Splines & Their Use in Computer Graphics – CS 779

This course deals with recently developed techniques for representing, manipulating and rendering "freeform" curves and surfaces assembled from polynomial curve segments or bivariate polynomial surface "patches" (typically cubic and bicubic). Applications of interest lie in documentation graphics, computer aided design (especially solid modeling) and synthetic image generation. CS 779 is intended to appeal both to those interested in numerical analysis and to those interested in computer graphics. A technical report written by the instructors [Bartels83a] serves as the principal text, although students also read a small number of papers.

Those taking the course should ideally have had an introductory exposure to both graphics and numerical analysis, but no student reasonably grounded in either should hesitate to take the course for lack of grounding in the other; preliminary material is covered wherever necessary.

This is a new course being given for the first time in the Fall of 1983. In the future it will be taught once yearly in a term during which CS 788 is not being offered. There are no programming assignments in the traditional sense, although fairly extensive use is made of a symbolic manipulation package (Vaxima or Maple).

Laboratory Hardware

Our principal graphics display, purchased in 1979 with funds provided by an NSERC equipment grant, is an Adage/Ikonas raster display system. The Ikonas consists of a 512×512 by 32-bit deep frame buffer feeding parallel 256-word red, green and blue colour maps via a 35-bit-in/34-bit-out cross bar switch, a high speed (200 ns) bit-slice microprocessor, micro-programmable matrix multiplication board, Motorola 68000 "multifunction peripheral controller," and real-time colour frame grabber. Microprograms may be written in assembly code or a subset of the C-language which allows the programmer to drop into assembly language to code time-critical inner loops. Host support is provided by a device driver for the C implementation of the CS 688 graphics package used in the Lab, and by a suite of special purpose commands.

Our second major piece of graphics hardware is an Evans & Sutherland Multi

Picture System, funded in 1982 by an NSERC strategic equipment grant. The MPS is a refresh vector display with transformation and clipping hardware, and equipped with a tablet, joystick, light pen, knobs and a function box. For many applications, such as surface modeling, the MPS is more appropriate for the highly interactive design of (wire-frame) images, which can then be rendered more slowly as solidly shaded objects on the Ikonas. Host support is provided by a graphics package for the E&S Picture System 2 which was installed and modified for the MPS by Carol Hayes as part of her masters work.

Our primary computer ("watcgl"), to which both the Ikonas and the MPS are connected, is a VAX 11/780 with eight megabytes of memory, two 256-megabyte removable disks, three 67-megabyte removable disks, and a triple density high speed tape drive. The VAX is connected to both the Gandalf and Sytek campus communication networks, and is also accessible via three 1200 baud dialup ports. The VAX runs Berkeley Unix and is used solely for research in computer graphics. It was purchased in 1982 with funds provided by an Ontario BILD grant to replace our PDP 11/45 and upgrade our original Ikonas RDS 2000 display to an RDS 3000.

The Ikonas is also interfaced to a Honeywell Level 6 minicomputer which we share with the Software Development Group. The Level 6 runs an operating system called Thoth, which was developed by the University of Waterloo Software Portability Group and is of particular interest to us because it encourages the use of multiple processes and message passing. By mid-1984 we expect to have moved the work currently being done on the Level 6 to the Motorola 68000 attached to our Ikonas; we are now in the process of installing a similar operating system (Harmony) developed by WM Gentleman at the National Research Council of Canada.

Other graphics equipment available in the Laboratory, purchased with NSERC and University of Waterloo grants, includes:

- several Summagraphics Bit-Pad graphics tablets;
- a GTCO Digi Pad 5 Pressure Pen Tablet;
- several optical mice;
- an intelligent trackball designed and developed by David Martindale as part of his masters essay;
- an HP 7221C 8-pen plotter;
- a Dunn 631 colour camera, which is connected to the video output of the Ikonas and generates colour 35mm slides and 8x10 Polaroid prints;
- two Sony 3/4" video tape recorders and editor;
- three HP 2648A graphics terminals;
- a Keynote Designs KD404 graphics terminal;

- two Electrohome Telidon terminals.

We will shortly be acquiring an IBM PC-XT, equipped with a Winchester disk, digital mouse, and network interface. It will run an operating system called Port developed by the Software Portability Group. We are currently looking for suitable graphics hardware with which to augment our Port workstation.

The Computer Graphics Laboratory is in the process of moving to a new home on the Sixth Floor of the Math and Computation Building after four years on the Fourth Floor. The move is the result of expansion by the EMS Library, which will be claiming all of the research space now located on the Fourth Floor. The new facilities will represent a modest increase in floorspace, both for the machine room and for personnel space. The Lab area houses all CGL equipment and is used by faculty and graduate students for most of their research work. Weekly meetings, film and video screenings, informal seminars, and day-to-day gatherings are held there as well.

Surface Modeling and Computer Aided Design

Surface modeling with splines is a major interest of Profs. Bartels and Beatty, who have been working closely with Prof. Brian Barsky from the University of California at Berkeley. Activity has centred on the use of B-splines and on the development of new Beta-spline techniques for computer aided design applications.

B-splines were introduced to computer graphics by Gordon and Riesenfeld in the mid-1970's. They provide a flexible and powerful means of smoothly assembling freeform parametric curves and surfaces from many polynomial "curve segments" or "surface patches," and are a substantial improvement over the techniques introduced earlier by Coons and Bezier. Unfortunately much of this power is often unused in computer graphics because the literature in B-spline mathematics is generally directed at numerical analysts and not at the general computer scientist; one of our objectives has been to make this material more readily accessible.

The Beta-spline curves and surfaces introduced recently by Barsky are a generalization of the uniform cubic B-splines in which "geometric continuity" is enforced at the juncture of segments and patches, instead of the parametric continuity which is required of B-splines. More flexible and intuitive control of curve and surface shape results.

Spline Mathematics

In 1981 the Lab was awarded an NSERC strategic operating grant for research on "2-D and 3-D Interactive Computer Graphics Modeling Systems." This work, supported also by a $24,000 contract with the National Research Council, led to new results in the mathematics of Beta-splines which were reported at the 1983

Siggraph Conference and published in *Transactions on Graphics* [Barsky82a, Barsky83a]. Both the B-spline and Beta-spline approaches provide means of defining a curve or surface as an approximation to a set of *control vertices* which the user specifies; movement of a control vertex changes the curve. The basic Beta-spline approach provides two additional global tension parameters (called β_1 and β_2) with which shape may be controlled. In the work referred to above, a technique was introduced for specifying distinct values of the shape parameters at each curve or surface joint in such a way as to retain geometric continuity while providing local control of tension.

A comparative study of shape control for B-spline and Beta-spline curves was presented at the 1983 Graphics Interface Conference [Barsky83b]. A comprehensive treatment of all the above material, together with a careful introduction to the mathematics of B-splines and Beta-splines (including a development of the "Oslo Algorithm" for control polygon/graph refinement) is provided by [Bartels83a]. This technical report is the course text for the seminar on the use of splines in computer graphics (CS 779) described earlier. A revised version will be supplied as notes for a tutorial presented by the authors at the 1984 Graphics Interface Conference, and will eventually be published as a monograph.

In addition to the results mentioned above, we have obtained a "divided difference" definition of the (cubic) Beta-splines which provides an alternative means of specifying distinct β values at each joint, and allows the parametric intervals corresponding to curve segments or surface patches to be of different lengths (*nonuniform knot spacing*). This definition of the Beta-splines is analogous to the traditional divided difference definition of the B-splines. A brief description appeared in [Bartels83b]. A more detailed treatment is in preparation.

Spline Software

We have been pursuing software implementations of this material in parallel with the mathematical analyses. Software for the rendering of B-spline and Beta-spline curves and surfaces was integrated into our standard Ikonas software by Leo Boutette as part of his masters. A Waterloo Co-op student, Ines Hardtke, has been designing interactive tools for experimenting with various spline techniques on our Multi Picture System. A second student, Carol Hayes, provided Hardtke with a package of routines for interfacing to the MPS as part of her masters work.

Complementary work on rendering is being pursued by Mike Sweeney, a graduate student. Mike is studying illumination models in conjunction with the use of a ray-tracing algorithm for visible surface computation (see Figures 2, 3 and 4). The primitive objects available are polygons, spheres, cylinders, fractal triangles and spline surfaces (rendered by subdivision). Objects are grouped (and in the case of splines, subdivided) into nested bounding boxes – if a ray misses the box, then it isn't tested against its contents. Diffuse, Phong, and Cook/Torrance shading models are available. Arbitrary images can be texture mapped onto polygons and spline surfaces. The objective is to extend and improve the efficiency of existing algorithms; the application of ray tracing to splines, for example, is new.

Geoff Sherwood and Richard Cole, recent graduate students, have been exploring more algorithmic methods of constructing surfaces. Sherwood's software is used to construct surfaces of revolution and extrusion. In the former case, the user may interactively specify a profile and a cross section – as the profile is rotated it is scaled (with respect to its distance from the axis of rotation) by the cross section. For surfaces of extrusion a cross section is swept along an arbitrary path. The usual shading rules are available, including a Cook/Torrance module supplied by Mike Sweeney. Figures 5 and 6 illustrate the kinds of objects and scenes which he is able to construct.

Richard Cole's work takes an entirely different approach. His software least-squares-fits a multinomial surface to a set of points provide by the user. Recurrence relations based on orthogonal multinomials make it possible to construct these surfaces quickly and efficiently. Since he is actually using parametric multinomials, it is necessary to fix upon a parametric coordinate system (cylindrical, the unit square, etc.) and to associate particular values of the independent parameters u and v with each data point. Techniques from statistical pattern recognition are used to accomplish this task.

Benesh Dance Notation

For two years now we have been working with Prof. Rhonda Ryman of the University of Waterloo Dance Department to develop an interactive editor for Benesh Dance Notation. Curiously, satisfactory written languages for recording movement have been developed only in this century; it is a sad fact that much dance history has been lost because it was recorded nowhere but in the minds of the choreographer and dancers. A difficulty both with Benesh and with Laban (the other commonly used notation) lies in the volume of data involved and the difficulty of editing and printing a score.

The Benesh editor implemented by Baldev Singh, then a graduate student, is the prototype of a system which we would eventually like to install at the National Ballet of Canada in Toronto. (See Figures 7 and 8.) Although our prototype runs on a VAX and generates output for our Ikonas display, we expect that it will soon be economically reasonable to implement our editor on a suitable workstation. We have applied to SSHRC for funds with which to study and extend our prototype, and to design such a workstation editor after having reviewed further usage of our editor by practising choreologists. With respect to a workstation environment, we are exploring the feasibility of using an IBM PC-XT based system and the Port operating system developed by the University of Waterloo Software Portability

Group.

Robyn Hughes (former choreologist for the National Ballet), Ingrid Filewood (present choreologist for the National Ballet), and Monica Parker of the Benesh Institute of Choreology in London have been consulting with us on this project.

We described our Benesh editor at the Tenth Annual Dance in Canada Conference in June 1982. Papers emphasizing the dance and man-machine interaction aspects of the editor have appeared in the CORD Dance Research Journal [Ryman83a] and at the 1983 Siggraph Conference [Singh83a], respectively. The Ontario Science Center is contracting with Human Computing Resources in Toronto to move our Benesh prototype onto OSC equipment for use in an exhibit during the Summer of 1984.

Colour

Because most of our work involves the use of a colour display, we have become interested in colour technology, the psychophysics of vision, the psychological aspects of colour usage, and in colour aesthetics. On the one hand we would like to apply knowledge from other fields to man-machine interaction problems; at the same time raster equipment offers an opportunity to conduct experiments which psychologists have heretofore found difficult to perform.

Results of our early studies have appeared in [Goetz82a, Beatty83a]. These represent an effort on our part to disseminate elementary information about the perception and use of colour which is not well known in the graphics community but is of great utility in using the low and medium cost colour displays and film recorders becoming widely available.

It is a remarkable fact that a number of seemingly elementary questions regarding colour are still open. One such question is simply, "what is the best way of selecting a colour?" We are studying two aspects of this question.

It is well-known that there are three degrees of freedom in specifying a colour; the range of all possible colours is appropriately represented by a volume in three space. We would like to know what set of axes are easiest to work with; red-green-blue and hue-saturation-value axes are the most common in computer graphics, and industry most often uses one of the CIE systems (XYZ, LAB or LUV), whereas commercial television uses the NTSC YIQ representation. Given any set of axes, we would also like to know what interaction technique is most easily used to alter coordinate values so as to select a colour.

We are therefore conducting a series of colour matching experiments in order to determine which of several coordinate systems and interaction techniques are most easily learned and used. Results from the first series of experiments will be used to guide further studies in which additional input devices, interaction techniques, coordinate systems and matching tasks will be examined. This work is being carried out by Michael Schwarz, a graduate student, in cooperation with Dr. William Cowan of the National Research Council and Dr. Jane Gentleman of Statistics Canada.

We have also actively supported Drs. William Cowan and Colin Ware (a PhD psychologist, previously at NRC and now a graduate student in the Lab) in preparing a one-day tutorial on colour for presentation at Waterloo and at the Siggraph '83 Conference.

Documentation Graphics

A major application of interest to us is that of "documentation graphics," in which one mixes text and graphics to produce visuals for oral presentations or illustrations for technical documents.

There are a variety of sources from which such graphic images may be obtained: digitized photographs; the graphical output from application programs; graphics languages; and interactive software systems for picture manipulation. We have chosen of late to concentrate our efforts on the interactive creation of images. The *Paint System* implemented in the Computer Graphics Laboratory by Eugene Fiume, Darlene Plebon and Richard Beach has provided a vehicle for the exploration of programming methodologies and user-interfacing techniques, in addition to yielding experience with picture creation *per se* [Plebon82a, Beach82a].

A Paint picture consists of an array (the *frame buffer*) of coloured picture elements (*pixels*) which are displayed on a colour monitor. A pointing device (a *graphics tablet* and *puck*) is used to locate a "paint brush" (an iconic *tracker*) on the monitor. When a button on the puck is depressed, Paint modifies image pixels with the current brush pattern. Several painting techniques have been implemented: copies of the brush may be rubber stamped; the brush pattern may be inked along the path traced by the tracker as the puck is moved across the tablet; the effect of an air brush can be produced by tinting pixels in a varied pattern about the brush.

Movement of the tracker can be constrained, say to horizontal or vertical lines, to obtain the "ruled" images more often needed in technical illustrations (Figure 9), or freehand images may be sketched (Figure 10). Paint is used by members of the Laboratory and others as a production facility for the preparation of 35mm slides, subject to availability of the Honeywell Level 6 on which Paint runs and contention for the Ikonas display. We have also arranged for various local artists (Mary Ann Mobley from Wilfred Laurier University, Dawn Woodey, and students in the Waterloo Fine Arts Department) to experiment with Paint, and we run an annual "Paint Contest" to get "novice-user-feedback." Finally, a subset of Paint was implemented on the Port Workstations by members of the Software Portability Group, in consultation with Darlene Plebon.

As useful as Paint is, it nonetheless has limitations. A Paint image is simply an array of pixels; Paint has no structural knowledge about the contents of an image. Our next objective is to make it possible for the user to work with a hierarchical object structure describing the image in front of him. In addition to facilitating image manipulation, such a structure is preferable when embedding images in documents or transmitting them across low speed lines. Darlene Plebon explored such techniques for structuring pictures in her masters thesis [Plebon82a].

Another member of the Laboratory (Rick Beach, a PhD student) is studying graphical style. Rick would like to develop a means of abstracting the graphical attributes of images so that they can be instantiated appropriately and automatically in a variety of media. This is analogous to the way in which text processing systems abstract such notions as title, author, section heading, paragraph, or journal reference and automatically generate typeset copy appropriate for a particular journal and output device. Of equal importance to Beach is the goal of obtaining consistently composed graphical images embodying the principles of good design followed by graphics designers and artists by embedding such "style rules" in a typesetting system. Yet another subject of great interest is the question of how best to implement automatic page layout of documents mixing graphics and text, and how interactive layout directives might interact with automatic layout to improve the result. Beach is pursuing this work at Xerox PARC, supervised jointly by faculty of the Laboratory and by staff at PARC. Preliminary results have been reported in [Beach82a].

One of our graduate students (Dave Forsey) is moving Paint from the Honeywell Level 6 to the Motorola 68000 attached to the Ikonas bus. The 68000 will run the Harmony operating system mentioned earlier. Harmony supports message passing and multiple process program structuring, which we need for Paint, as well as multiple processor program architectures (areas of continuing interest to us – see [Booth82a], for instance). For a number of reasons it will be necessary to move Paint before we can enhance it.

Firmware

The user of a vector display is manipulating lines, which can be done quite rapidly, while the user of a raster display is generally modifying areas on the screen. Performing these computations on a host computer and communicating new intensity values for tens or hundreds of thousands of pixels is a lengthy process. It is for this reason that one often arranges for special hardware, or a separate high-speed cpu like the Ikonas bit-slice micro-processor, to have direct access to the frame buffer.

The Ikonas Bit-slice

Using the Ikonas bit-slice effectively requires substantial effort since the architecture is highly parallel and computation is controlled by 64-bit-wide micro-instructions of considerable complexity. To facilitate debugging of bit-slice code, Paul Breslin (then a masters student) provided us with a simulator for the Ikonas micro-processor so that programmers can debug their code interactively in a totally controlled environment before moving code to the micro-processor itself.

Using the simulator, Breslin implemented a firmware display processor on the micro. Nested segments, picking, highlighting and dragging (using the writemask and autoclear features of the image memory and frame buffer controller, respectively) are available. The basic primitives are lines (which may be anti-aliased), rectangles, and triangles. A host interface to the display processor triangulates arbitrary convex polygons, which may be constant, Gouraud or Phong shaded. This package has been used in a number of other projects, including the Benesh editor, VLSI editor, and two spline manipulation packages.

Breslin's work was substantially complete before the Lab acquired a matrix board for the Ikonas, and so his display processor does not provide for embedded transformations. A second graduate student (Al Paeth) is undertaking extensions of Breslin's package to remedy this need. We are considering other modifications which Breslin suggested in reviewing his work, including the use of high precision coordinates to facilitate anti-aliasing and the anti-aliasing of triangles. Because the matrix board is actually a programmable micro-processor and is suitable for subdivision and forward differencing, we plan to add spline primitives to the display processor as well.

Since writing correct micro-programs is a tedious and error-prone activity, another masters student (Preston Gurd) implemented a compiler which translates a subset of C into micro-code for the bit-slice. The notion is that initialization code can be written in a higher level language so that the micro-programmer can concentrate on directly micro-coding the inner loops of an algorithm, if that is necessary. (The portions of our Paint program which were moved to the bit-slice for reasons of speed did not need such hand optimization.) This compiler was described at the 1983 Sigmicro Conference [Gurd83a].

Since completing his masters, Preston has retargeted his compiler for the bit-slice in a M68000-based workstation designed by Electrohome. He is working on a similar project for Datacube. Preston has also written a completely new assembler for the Ikonas bit-slice which is an order of magnitude faster than the assembler supplied by Adage/Ikonas.

The CGL Trackball

Our experience with various tablets and "mice" has led us to conclude that the vendors of graphics input equipment have not taken full advantage of the flexibility to be had from the micro-processors commonly embedded in such devices. David Martindale did a comparative study of several input devices and designed, built and programmed a more intelligent trackball/dial/button assembly.

Its communications protocol is designed for easy use, and it may be configured to report trackball or dial changes at fixed intervals, whenever a change exceeds a specified threshold, or when either a specified time has elapsed or a specified threshold movement has occurred. If desired, data may be queued within the trackball to avoid data loss due to most overload conditions. The trackball is now undergoing user testing and a paper describing its design is in preparation.

Scan Conversion and Anti-Aliasing

A particular interest of Prof. Field's has been the design and analysis of fast algorithms for rendering anti-aliased lines, circles, rings, ellipses, and triangles [Field82a, Field82b, Field83a, Field83b]. A characteristic of the algorithms he has developed is that they do not use floating point, multiplication, or table lookup operations within their inner loops. Thus they are ideal for hardware or firmware implementation. Figure 11 illustrates the differences between several bilevel (not anti-aliased) and anti-aliased objects rendered by algorithms Field has developed. (Figure 12 illustrates the anti-aliasing of text.)

Prof. Field is currently designing a VLSI circuit to implement a bilevel line algorithm which will form the controlling logic for a circuit to draw anti-aliased lines. It is expected that plot times of less than 200 ns per pixel will be achieved. A major component of the bilevel circuit has been sent for fabrication (see Figure 13); testing should begin within the next few months.

Preliminary work on a second bilevel line algorithm indicates that hardware implementation will achieve point plot times of less than 30 ns. Further work in this area will concentrate on the design and analysis of scan conversion algorithms for other primitive geometric objects and their eventual implementation. The Ikonas 3000 provides a convenient environment in which these algorithms (and the circuit mentioned in the preceding paragraph) can be tested.

A major goal of this research is to obtain viable alternatives to traditional scan-line-ordered image generation. The scan conversion techniques we develop will relieve much of the burden on the host machine. A consequence of this approach is that algorithms in the graphics pipeline, such as hidden surface analysis and shading, must now operate at the level of primitive objects. We are studying the applicability of techniques recently developed in the field of computational geometry to solutions for this problem.

The Tektronix Geometry Processor

The Geometry Processor is a high performance TTL micro-processor whose hardware and firmware have been designed to rapidly perform the matrix computations commonly needed in two and three dimensional graphics. The GP was designed and built by Tektronix, Inc. Profs. Beatty and Booth, as consultants, participated in the development of its macro instruction set, of a high level assembler in which the firmware implementing that instruction set was implemented, and of a host graphics package for interfacing with the processor [Beck80a, Booth81a, Beatty81c].

Tektronix has given this hardware to the Computer Graphics Laboratory. Steve Hayman (a graduate student) is moving the software, which was written for an older version of Unix running on a PDP 11/44, to our VAX. We are interfacing the GP to the VAX Unibus and to the Ikonas internal bus so that we can use the Ikonas bit-slice to scan convert polygons which have been transformed and clipped by the GP. The GP's firmware will be augmented with macro-instructions for subdividing B-spline and Beta-spline surfaces to form polygonal approximations which will then be scan converted by the Ikonas bit-slice. Eventually the Ikonas microprocessor will be augmented by custom scan conversion hardware designed by Prof. Field.

Future Projects

The Graphics Group at the National Research Council is purchasing an Ikonas display system similar to ours, and we are pursuing a contract with them for the development of polygon manipulation and scan conversion firmware which they will use in a task planning system for robotic assembly.

We would also like to explore the use of parallel algorithms to speed up display by placing a number of bit-slice and M68000 processors on the Ikonas' internal bus. This is one of the projects for which we will be using the Harmony operating system; processors will communicate via Harmony's message passing primitives.

Computer Animation

The Laboratory's work in animation is being pursued jointly with the French Animation Studio at the National Film Board. Doris Kochanek, now employed at the Film Board, explored techniques for improving the computer-based animation being done there while working on her masters in the Laboratory. Kochanek concentrated most heavily on the problem of smooth inbetweening for keyframe animation. She developed and implemented a technique in which Hermite interpolation is used to smooth both the path followed by an animated object, and changes in its velocity. In a later generalization of this technique she has found a way to translate intuitively simple ideas of curve bias, tension and "continuity" into the tangent vectors from which Hermite interpolating curves are computed, further simplifying the process of inbetweening. She has also experimented with the combined use of vector and raster graphics, using a bi-pack camera, to eliminate rastering in computer generated animation. A paper describing her work is now in preparation; a preliminary description appears in [Kochanek82a].

The NFB is in the process of acquiring a computer to support future animation work. Prof. Booth and David Martindale (CGL lab manager) are working with Robert Forget (Director of French Animation) and Kochanek in configuring equipment. Prof. Booth is planning to spend his sabbatical (1984-1985) developing animation tools at the Film Board. Several of our students will spend time there with him. Terry Higgins, a Waterloo undergraduate who will begin graduate work with us in January of 1984, is presently at the National Research Council working under contract for the National Film Board on paint software as part of this project. He is incorporating some features of our Paint program, as well as drawing on previous experience with the graphics group at NRC as a Co-op student.

Related work on multi-plane animation is being done by Rob Krieger. At present he is implementing tools for the manipulation of raster images, following work presented at Siggraph '80 by Catmull and Smith. These tools will enable him to rotate, skew, or re-project raster images through an anti-aliasing filter and back into the frame buffer. Among other things, he will then be able to map digitized or synthetic texture patterns onto arbitrary digital cels of an animation frame. (See Figure 14.)

Other Research Areas

Interactive Seating Layout

One of our graduate students (Mert Cramer) is taking advantage of an extensive background in theatre management to develop a tool for the interactive design of seating plans for theatres and auditoriums. Such design has previously been done in a strictly *ad hoc* way. Cramer is developing software that will allow him to specify a probability distribution for the height of viewers and then interactively evaluate and modify a seating layout. The user will be able to: determine the probability that the stage will be visible from a given seat; given a height for the back row, establish a seating contour which yields a constant probability that the stage will be visible from every seat; maximize the probability of the stage being visible given a specified probability, size constraint or bias for some portion of the audience.

This work is being pursued in cooperation with Prof. William Crocken from the Department of Theatre and Film at The Pennsylvania State University.

Interactive Graphical Graphics Debugging

An obvious area of application for the graphical interaction techniques and ergonomic principles in which we are interested is the debugging of programs which embody these techniques and principles. By their very nature, graphics programs are likely to make use of complex linked data structures and list processing algorithms whose correctness is difficult to verify. Consequently we are exploring the design of interactive debugging systems specifically tailored for the graphics environment.

This topic is being pursued by Edward Bulman, a PhD student at Waterloo and faculty member at Laval University. His goal is to design and implement a system, in a Pascal environment, for graphically displaying the dynamic behaviour of running programs. The immediate goal is to explore ways of visualizing program behaviour interactively. The long range motivation, of course, is to provide ourselves with a general purpose tool which we can use elsewhere in the Laboratory in developing graphics algorithms.

Interactive Stick Layout for VLSI

Kevin Szabo, a masters student in the Electrical Engineering Department, used CGL facilities to design an interactive program on our Ikonas for creating and editing bipolar VLSI circuits. (See Figure 15.) The user manipulates a "stick" representation developed by Prof. Elmasry of the EE Department. Upon request Szabo's software will convert the stick representation to a mask layout; he is currently designing an algorithm for automatically compacting these masks, subject to specified design rules. Szabo's software makes extensive use of Breslin's bit-slice firmware and the menu package developed by Singh for the Benesh editor.

Lab Organization

Members of the Lab meet weekly from 1:30-2:30 on Wednesday afternoons. Each meeting begins with a half hour technical presentation, followed by a discussion of laboratory problems requiring attention ("action items") and a report on the resolution of previously discussed problems. The faculty reports on such perennially interesting topics as funding, maintenance and space. The chair is rotated in alphabetic order among Lab members, and everyone is expected to attend. Meetings are held strictly to one hour.

The Laboratory as a whole meets periodically with the University of Toronto graphics group to exchange ideas and report on research. The venue alternates between Toronto and Waterloo.

In addition to a full time lab manager and part-time administrative assistant, a "CGL Unix teaching assistantship" is available through the Math Faculty. This position is rotated among two or three people in various stages of expertise. The Unix TA is expected to attend to file system backups, and serves as a consultant to other members of the Lab. An important part of the job is to become familiar with the organization of the operating system's entrails. Experienced TA's are expected to spend some time bringing novice TA's up to speed.

A research assistantship is also allocated for a "demo czar" who conducts demonstrations of Lab software and tours of our facility. The demo czar is responsible for maintaining the support software comprising the demo shell and for installing newly completed demonstration programs.

Other members of the Lab customarily volunteer on a round-robin basis to attend to smaller and more transitory tasks.

Lab Policies

Students interested in joining the Graphics Laboratory are expected to have taken the Introductory Graphics Class. Students who have satisfied this prerequisite are encouraged to consider a masters essay, masters thesis or PhD research in graphics, as appropriate.

For those students who require financial support, two teaching assistantships are available each term for CS 488/688. A number of research assistantships are also available. From time to time the Lab also undertakes a limited amount of contract work, which is ordinarily used to support graduate students.

Every member of the Laboratory is expected to perform a reasonable share of the miscellaneous work necessary to keep things running smoothly, to help integrate new members into the Lab, and to document the work they do so that it is readily usable by others and can be easily maintained after their departure.

For students making satisfactory progress, the Laboratory contributes towards the cost of attending the annual Siggraph Conference, or of other relevant travel. Such students are expected to give a technical presentation based on their trip at a subsequent Laboratory meeting.

Students may obtain information about joining the Laboratory and financial support, subjects of a separate document, from any of the CGL faculty members.

Contract Work

In 1982-83 the Laboratory had a $24,000 contract with the National Research Council for which we surveyed the use of splines for surface modeling. The emphasis was on extracting results from the numerical analysis literature and casting it in a form appropriate for computer scientists. The fruit of this labor was a lengthy report [Bartels83a] and a one-day tutorial at NRC.

The Laboratory recently completed a $30,000 development contract for Management and Planning Software Group of Toronto. Software for a basic device-independent graphics package was designed and implemented for MPS, and a graphics module was then designed and implemented for their Interactive Decision Support Software system using that package. There are presently device drivers for the Hewlett-Packard 7221 8-pen plotter, line printer, and Electrohome Telidon terminal. The package itself is device independent, and additional device drivers can easily be added as necessary. The IDSS module supports basic business graphics including line charts, bar charts and pie charts, with solid fill and cross-hatching. (See Figure 16.)

Outside Activities

From 1977-1980 one or both of Profs. Beatty and Booth conducted a two day tutorial on Introductory Computer Graphics at the annual conference of the ACM Special Interest Group on Graphics (Siggraph). Booth served on the annual Siggraph Conference's program committee from 1979 to 1982, as did Beatty in 1981 and 1982. In 1983 they co-chaired the Conference.

CGL faculty members have also given one or two day tutorials on computer graphics for: Burroughs Corporation in 1979; IEEE Compcon in 1979; IEEE Tutorial Week West in 1979; Imperial Oil in 1980; Tektronix in 1980; Tutorial Week in Europe 1982; Graphics Interface Conference in 1982; the Santa Cruz Summer Institute in 1982 and 1983; the Waterloo Institute for Computer Research in 1982 and 1983; IEEE COMPSAC 1983. Most of the tutorial fees received for this work have been donated to the Laboratory. Beatty and Booth prepared a volume of journal reprints for the IEEE tutorials [Beatty82a].

Profs. Bartels and Beatty will be presenting a tutorial on the use of splines in computer graphics at Graphics Interface '84 in Ottawa.

Members of the Laboratory handled local arrangements for the Seventh Conference of the Canadian Man-Computer Communications Society, which was held at the University of Waterloo in June of 1981, and served on the program committee.

Profs. Beatty and Booth gave a half-day presentation entitled "Computer Graphics – What's Happening" at the 1980 Conference of the Canadian Society for Computational Studies of Intelligence. Prof. Beatty gave an invited talk on

"Colour in Computer Graphics" at the May 1980 meeting of the Statistical Society of Canada. An expanded version of the latter talk appeared in the *American Statistician* [Beatty83a].

The Future

With respect to teaching, our immediate plans are to provide practical experience with graphics tablets and "mice" for students taking the introductory graphics class, and to replace the Electrohome Telidon terminals now in use with Keynote Design KD404's. (This work is being done by Klaus Ahlers and Mike Sweeney.) Both the Topics class (CS 788) and the Splines class (CS 779) will be taught once each year. We intend to move CS 788 to the Winter term; CS 779 would then be taught one of the remaining terms.

We expect the Laboratory to grow to about 25 students by the end of 1984, and we intend that it remain at this level as we do not feel that four faculty members can supervise a larger number of students, or that the graphics VAX would be able to adequately support a larger group. We do intend to shift the balance between masters and doctoral students somewhat more towards the latter, so as to provide a more stable and continuous laboratory community.

Our chief hardware need at the moment is a second raster display, owing to the number of people now working in the Laboratory, the need which students developing interactive systems have for extended use of the Ikonas, and the desire by various people outside the Lab to make production use of our software for slide preparation. We expect contention for the Ikonas to worsen as the Lab grows and as hardware development projects periodically necessitate making our present Ikonas unavailable for periods of extensive testing. We have applied to NSERC for partial funding of a second Ikonas, to be supplemented by monies generated by CGL faculty consulting donated to the Lab.

Reliability of the two 67-megabyte Ampex disks inherited from the PDP 11/45 has also been a persistent problem. We will likely want to replace them with an RM05 or equivalent Winchester in the near future.

Harmony should be running on our M68000 by the Spring of 1984, at which time we will no longer have need of the Honeywell Level 6. We expect that the VAX will remain our principle computer until approximately 1986, at which point the shift to a multiple-workstation environment will be well underway. By then the cost of workstations with graphic capability sufficient for our needs should be reasonable, networking hardware and software should be well stabilized, and workstation environments should be rich, reliable and in common use. In the interim we will gain experience with such environments via work on our Motorola 68000 and IBM/Port PC. We are particularly interested in pursuing the use of message passing and multiple processes; our experience with Paint has convinced us that their use can substantially improve program structure [Beach82a, Booth82a].

The graphics package through which we access our Ikonas was a rather hurried re-implementation in C of our Pascal teaching package. It has evolved somewhat haphazardly in the last few years. An incoming masters student (Mike Herman) will be redesigning this package for his thesis. Some work along these lines has already been accomplished by Sylvia Lea, who redesigned and augmented our "scene description language" for her masters thesis, and as an undergraduate Mike Herman examined ways of maintaining configurable source code for the graphics package [Cargill81a].

We expect to continue placing substantial research effort on the modeling of freeform curves and surfaces and on allied rendering algorithms. Our major thrust in the next 2-3 years, however, will lie in the realm of interfacing psychology, visual perception, and interaction techniques. The comparative study of colour specification systems mentioned earlier is a first step along this path. We will be using the Benesh and Paint applications as testbeds for further analysis and evaluation of man-machine ergonomics.

Acknowledgements

The assistance of Dave Forsey and Nancy Greene in preparing the illustrations for this report is much appreciated.

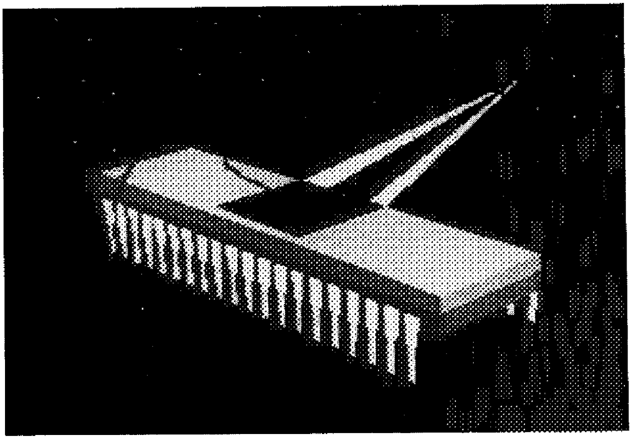

Figure 1. *Le Chip*, created by Stanley Yip and J. Sew for CS 488/688. This image was displayed on a Telidon terminal having 256×192 resolution and photographed with a Dunn 631 camera.

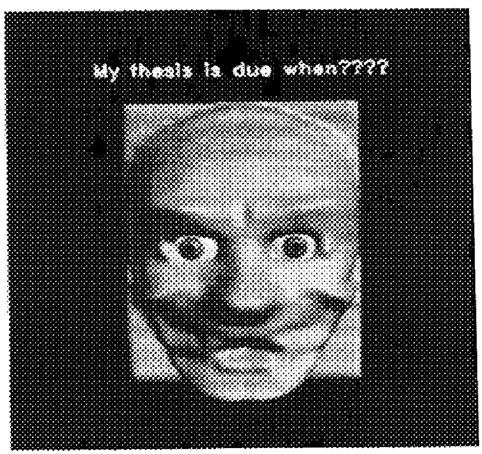

Figure 2. This is a bicubic B-spline surface, rendered by a combination of subdivision and raytracing. (Mike Sweeney.)

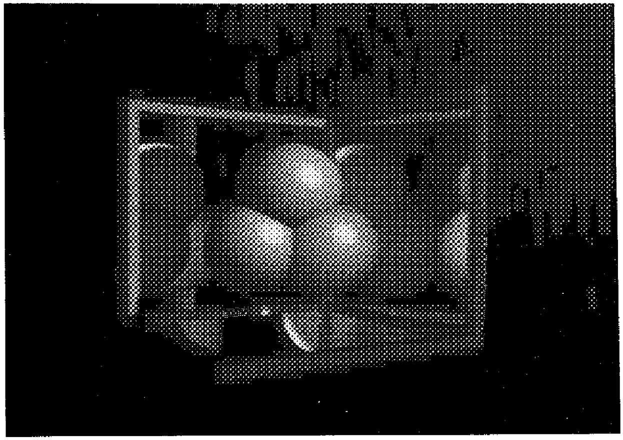

Figure 3. Three intersecting spheres are shown reflected in three planar mirrors. (Mike Sweeney.)

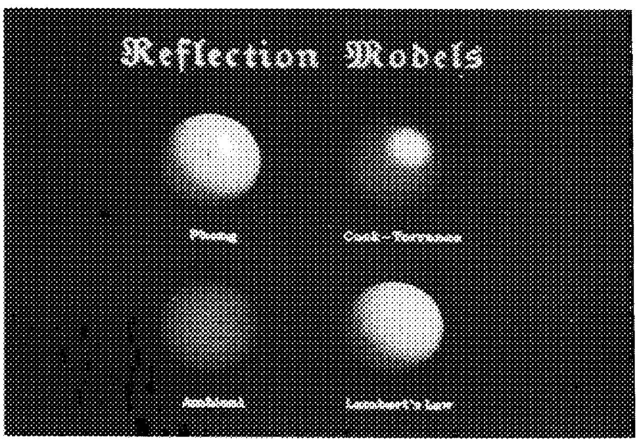

Figure 4. This slide illustrates the illumination models used in the Graphics Laboratory. A single light source is located upper right behind the reader. (Dave Forsey and Mike Sweeney.)

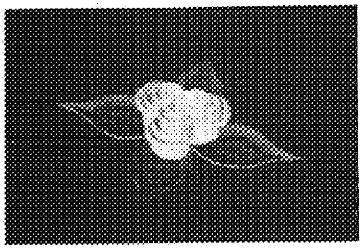

Figure 5. Surfaces of rotation (Geoff Sherwood).

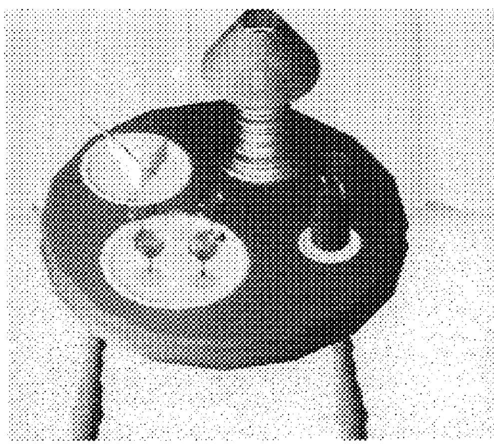

Figure 6. The curtain is formed by extrusion. The table and most of the objects on it are surfaces of rotation. The entire ensemble is rendered by a scan line z-buffer visible surface algorithm. (Geoff Sherwood.)

Figure 7. The Ikonas display screen, with a tablet and puck positioned in front of it. Movement of the puck across the tablet causes a programmable tracker to move across the display. At any given moment the form of the tracker and the menu items shown in reverse video indicate the state of the editor. The screen shows two lines of Benesh notation at the top and the frame being edited, together with several menus, at the bottom.

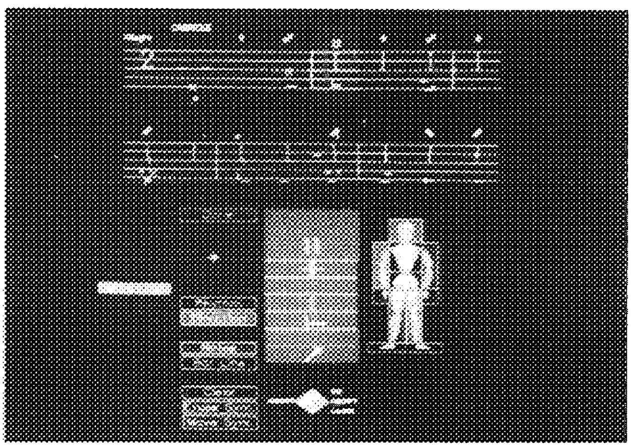

Figure 8. A closeup view of the display screen while the Benesh editor is executing. Note the use of a stylized human figure as an iconic menu (to the right of the working frame) for selecting body parts.

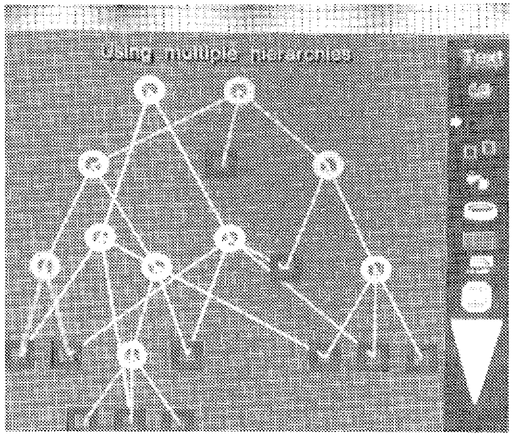

Figure 9. A view of the display screen while Paint is being run. The menu appears at the right. Each item in the menu is an icon which visually suggests the meaning of the corresponding command. A colour palette appears at the top. The image shown here is a typical technical illustration. (Prof. FW Tompa.)

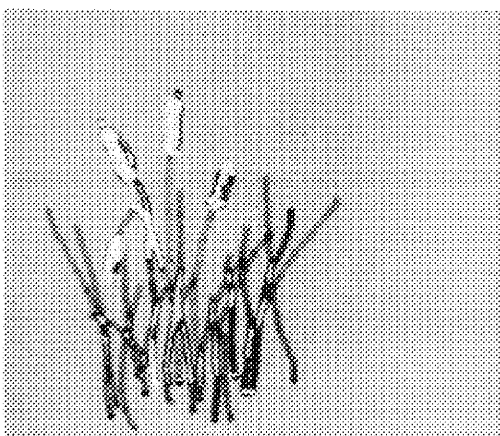

Figure 10. A Paint image (bullrushes) created by freehand sketching. (Sylvia Lea.)

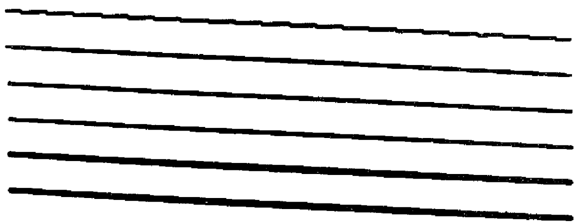

Figure 11. A comparison of bi-level and anti-aliased lines at 256×256 resolution. The top line shows the "staircasing" typical of aliasing. The lower five lines illustrate the use of different anti-aliasing filters to ameliorate these artifacts. 256 grey levels were used to smooth the jagged edges, apparent in the top line, which result from sampling an image discretely.

Figure 12. Anti-aliased text. These two letters from a raster display have been greatly magnified to show the intensity variations which are used to smooth curved lines and edges. Note the drop-shadowing.

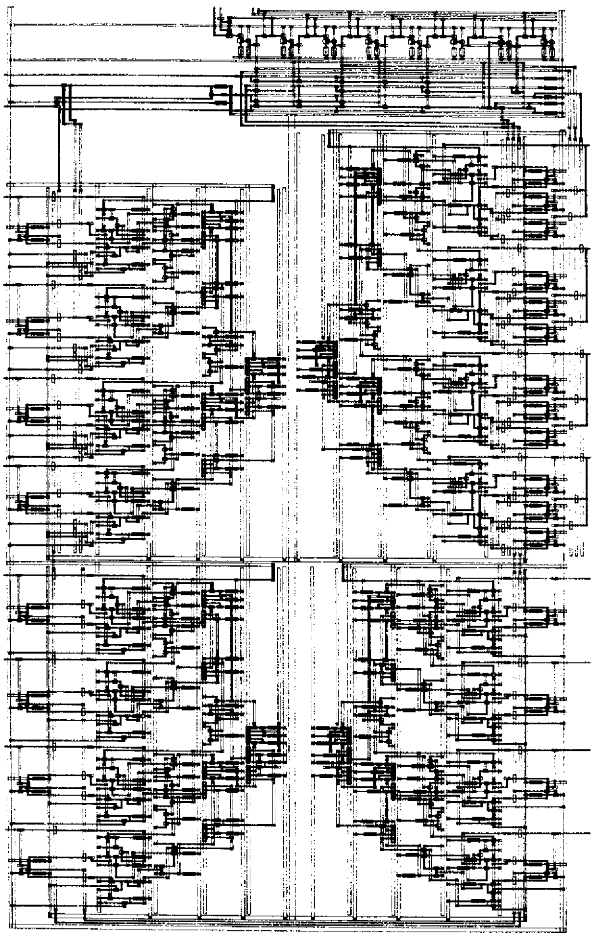

Figure 13. Part of a VLSI circuit for drawing anti-aliased lines. The cell depicted implements Bresenham's algorithm. Layout of the cell was accomplished with the CLAY silicon compiler.

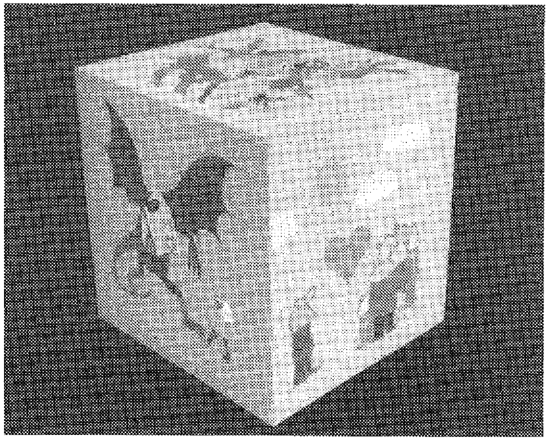

Figure 14. A variety of raster images projected onto the face of a cube. (Rob Krieger.)

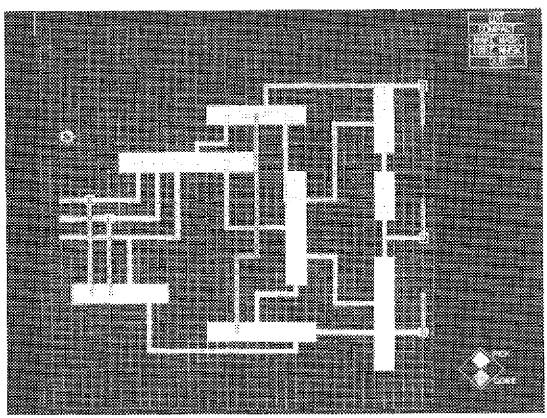

Figure 15. This picture shows a simple three input ttl NAND gate. The masks were generated automatically from an interactively designed schematic, the notation for which was developed by Prof. MI Elmasry (Director of the Waterloo VLSI Group). The notation carries both schematic (connectivity) and topological (placement and routing) information. The editor and mask generator are currently being moved to a stand alone work station. (Kevin Szabo.)

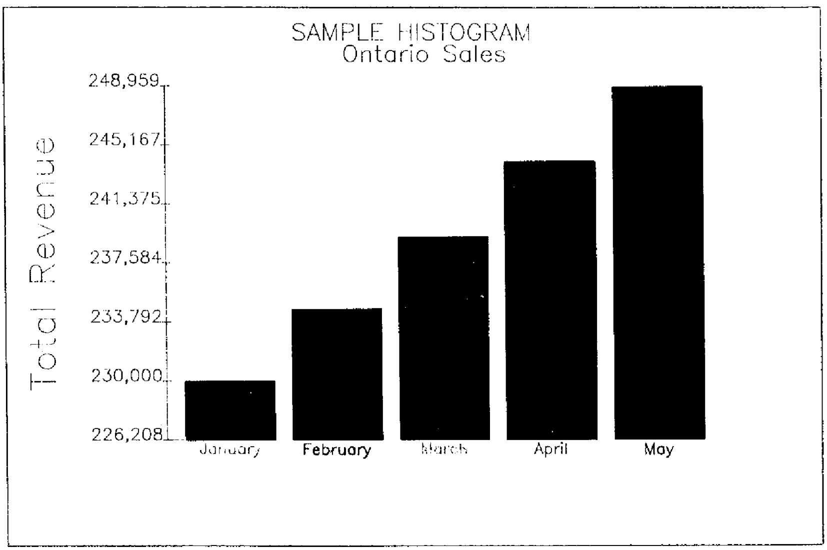

Figure 13. This figure was generated by the Management and Planning Software Group in Toronto on a Hewlett Packard 7221 pen plotter, using the graphics package implemented with them by the Computer Graphics Laboratory.

CGL Graduate Essays & Theses

1980

Joeseph H. Buccino (MA, 1980) [Electrohome]
A Reliable Typesetting System for Waterloo

Harry Yuen (MA, 1980)
Pascal DNI - A Program for Shading Molecular Models

1981

Gregory A. Hill (MA, 1981) [Mitre Corporation]
Antialiasing in Computer Graphics

Larry H. Matthies (MA, 1981) [Carnegie-Mellon PhD program]
Plane Sweep Algorithms in Computer Graphics

Robert Stephenson (MA, 1981) [Prudential Assurance]
Interactive Graphic Analysis for Multivariate Data

Alex R. White (MA, 1981) [University of Waterloo, MFCF]
Pic - A C-based Illustration Language

1982

Rodrigo Vicente Pablo Alliende (MA, 1982) [University of Chile]
Antialiased DNI

Leo G. Boutette (MA, 1982) [Ministry of Transport]
A B-Spline and Beta-Spline Curve and Surface Package

Paul H. Breslin (MA, 1982) [Human Computing Resources]
A Powerful Interface to a High-Performance Raster Graphics System

Susan M. Goetz-Obermeyer (MA, 1982) [AES Ltd.]
Colour Principles and Experience for Computer Graphics

Cathy Johnson (MA, 1982)
DNIF77 - A program for Shading Molecular Models

Doris H. U. Kochanek (MA, 1982) [National Film Board]
A Computer System for Smooth Keyframe Animation

Stephen A. MacKay (MA, 1982) [National Research Council]
Techniques for Frame Buffer Animation

Darlene A. Plebon (MA, 1982) [National Research Council]
Picture Creation Systems

Baldev Singh (MA, 1982) [Schlumberger-Doll Research]
A Computerized Editor for Benesh Movement Notation

1983

Richard H. Cole (MA, 1983)
The Use of N-Dimensional Curve Fitting in Computer Graphics

R. Preston Gurd (MA, 1983) [Independent Consultant]
A Microcode-C Compiler for the Ikonas Graphics System

Carol J. Hayes (MA, 1983) [Carleton Univesity]
System Support for an Evans & Sutherland Multi Picture System

Sylvia C. Lea (MA, 1983) [Bell Northern Research]
A Scene Description Language

Geoff C. Sherwood (MA, 1983) [Bell Labs, Denver]
An Object Maker System

Computer Graphics Seminars

1982

June 25: Fred Krull (General Motors Research Laboratory), *Computer Science Research at General Motors.*

September 20: Marceli Wein (National Research Council), *Interaction Techniques.*

September 28: Bill Buxton (University of Toronton), *Interaction Techniques.*

October 6: Baldev Singh (Schlumberger-Doll Research), *A Graphics Editor for Benesh Movement Notation.*

October 18: Bill Cowan (National Research Council), *The Psychophysics of Colour.*

October 20: Richard Beach (University of Waterloo & Xerox PARC), *Graphical Style - Towards High Quality Illustrations.*

November 1: Sue Goetz-Obermeyer (AES Ltd.), *Colour Principles for Computer Graphics.*

November 3: Al Paeth (Xerox PARC), *PIXELPLANES - Adding a New Dimension to VLSI-based Computer Graphics.*

November 22: John Dill (Cornell University), *Computer Graphics in the Engineering Curriculum at Cornell University.*

November 22: Doris Kochanek (National Film Board), *Computer Animation.*

November 24: Rob Pike (Bell Labs), *The Blit: Merging Bitmap Graphics and Unix.*

December 1: Bill Reeves (Lucas Film), *Computing at Lucasfilm Ltd - Darth Vader Meets the Computer Revolution.*

1983

February 21: Dan Field (Princeton University), *Algorithms for Drawing Simple Geometric Objects on Raster Devices.*

April 28: Aaron Marcus (Lawrence Berkeley Laboratory), *The Three Faces of the Future: Graphics Design and Computer Graphics.*

July 9: Bill Cowan & Colin Ware (National Research Council), *A Tutorial on Colour Perception* (full day seminar).

September 22: Tom DeFanti (University of Illinois at Chicago), *The Impact of Video Game Technology on Postverbal Communication.*

October 5: Doris Kochanek (National Film Board), *Computer Animation.*

October 11: Nick England (Adage Corp.), *New Developments & Trends in Raster Systems.*

October 19: Bill Cowan (National Research Council), *How to Talk About Colour.*

October 26: Colin Ware (University of Waterloo), *Colour Properties of the Environment.*

November 2: Marceli Wein (National Research Council), *Interactive Methodology.*

November 9: Ron Baecker (Human Computing Resources), *Interaction Techniques.*

November 10: Norm Badler (University of Pennsylvania), *Understanding Human Movement.*

November 16: Colin Ware (University of Waterloo), *Perception of Colour in the Environment.*

November 23: Bill Cowan (National Research Council), *Colour and Attention.*

November 30: Bill Buxton (University of Toronto), *User Interface Management Systems.*

These talks were organized by the Computer Graphics Laboratory, which defrayed expenses. Many were held in conjunction with CS 788.

References

[Barsky82a] Brian A. Barsky and John C. Beatty (1982), Varying the Betas in Beta-Splines, CR-82-49, Department of Computer Science, University of Waterloo, Waterloo, Ontario, 42 pages [also available as TR CSD-82-112 from the Computer Science Division of the University of California at Berkeley].

[Barsky83a] Brian A. Barsky and John C. Beatty (1983), Local Control of Bias and Tension in Beta-splines, *Transactions on Graphics* **2**(2), April, 27-52 [also available in the proceedings of the 1983 Siggraph Conference, which are the July 1983 issue of the Computer Graphics Quarterly, 17:3 pp 193-218].

[Barsky83b] Brian A. Barsky and John C. Beatty (1983), Controlling the Shape of Parametric B-spline and Beta-spline Curves, *Graphics Interface '83*, May, 223-232.

[Bartels83a] Richard H. Bartels, John C. Beatty, and Brian A. Barsky (1983), An Introduction to the Use of Splines in Computer Graphics, CS-83-09, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1, 182 pages.

[Bartels83b] Richard H. Bartels (1983), Splines in Interactive Computer Graphics, *Proceedings of the 1983 Dundee Conference on Numerical Analysis*, 26 pages [invited paper].

[Beach82a] Richard J. Beach, John C. Beatty, Kellogg S. Booth, Darlene A. Plebon, and Eugene L. Fiume (1982), The Message is the Medium: Multiprocess Structuring of an Interactive Paint Program, *Computer Graphics* **16**(3), July, 277-287 [proceedings of the 1982 Siggraph Conference].

[Beatty81a] John C. Beatty and Kellogg S. Booth (1981), Computer Graphics Laboratory - Fall 1981 Review, CS-81-38, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

[Beatty81b] John C. Beatty and Kellogg S. Booth (1981), Teaching Computer Graphics at the University of Waterloo, *Proceedings of the 1981 Conference of the American Society for Engineering Education*, June, 224-230 [invited paper].

[Beatty81c] John C. Beatty (1981), A User Interface for the CRL Geometry Processor, CR-81-88, Applied Research Group, Tektronix Laboratories, Beaverton, Oregon.

[Beatty82a] John C. Beatty and Kellogg S. Booth (1982, eds.), *Tutorial: Computer Graphics*, IEEE Computer Society [revised edition, Order Number 425].

[Beatty83a] John C. Beatty (1983), Raster Graphics and Color, *The American Statistician* **37**(1), February, 60-75.

[Beck80a] Jay Beck and John C. Beatty (1980), Special Purpose Geometry Processor Functional Specifications, CR-80-14, Applied Research Group, Tektronix Laboratories, Beaverton, Oregon.

[Booth81a] Kellogg S. Booth and Marc Wells (1981), The Special Purpose Geometry Processor Microcode Assembler, CR-81-11, Applied Research Group, Tektronix Laboratories, Beaverton, Oregon.

[Booth82a] Kellogg S. Booth and W. Morven Gentleman, (1982), Anthropomorphic Programming, *Conference on Language Issues for Large-Scale Computation*, March, Gleneden Beach, Oregon.

[Cargill81a] Thomas A. Cargill and Michael W. Herman (1981), Pinc: A Tool for Maintaining Configurable Software in Pascal, *Proceedings of the Fifth International Conference on Computer Software and Applications*, 428-432 [IEEE Computer Society].

[Field82a] Daniel E. Field (1982), Drawing Anti-aliased Lines, TR-309, Computer Science Department, Princeton University, Princeton, New Jersey, December.

[Field82b] Daniel E. Field (1982), Algorithms for Drawing Lines, Curves, and Triangles, TR-310, Computer Science Department, Princeton University, Princeton, New Jersey, December.

[Field83a] Daniel E. Field (1983), Algorithms for Drawing Simple Geometric Shapes on Raster Devices, Ph.D. dissertation, Princeton University, Princeton, New Jersey, June.

[Field83b] Daniel E. Field (1983), Two Algorithms for Drawing Anti-aliased Lines, Department of Computer Science, University of Waterloo, Waterloo, Ontario [submitted to *Graphics Interface '84*].

[Goetz82a] Susan M. Goetz, John C. Beatty, and Deryl J. Rasquinha (1982), Colour Principles and Experience for Computer Graphics, *Graphics Interface '82*, May, 313-322.

[Gurd83a] R. Preston Gurd (1983), Experience Developing Microcode Using High Level Language, *Proceedings of the 1983 Sigmicro Confer-*

ence.

[Kochanek82a] Doris H. U. Kochanek, Richard Bartels, and Kellogg S. Booth (1982), A Computer System for Smooth Keyframe Animation, CS-82-42, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1, 84 pages.

[Plebon82a] Darlene A. Plebon and Kellogg S. Booth (1982), Interactive Picture Creation Systems, CS-82-46, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1, 65 pages.

[Ryman83a] Rhonda S. Ryman, Baldev Singh, John C. Beatty, and Kellogg S. Booth (1983), A Computerized Editor for Benesh Movement Notation, *CORD Dance Research Journal* [to appear].

[Singh83a] Baldev Singh, John C. Beatty, Kellogg S. Booth, and Rhonda Ryman (1983), A Graphics Editor for Benesh Movement Notation, *Computer Graphics* **17**(3), July, 51-62 [proceedings of the 1983 Siggraph Conference].