PROCEEDINGS

6th symposium on small computers in the arts

OCTOBER 10–12, 1986
PHILADELPHIA, PENNSYLVANIA

SPONSORED BY:
Computer Society of the IEEE
Small Computers in the Arts Network, Inc.

in cooperation with:
ACM/SIGGRAPH
Delaware Valley Chapter ACM/SIGGRAPH

IEEE Computer Society Order Number 755
Library of Congress Number 86-45867
IEEE Catalog Number 86TH0164-4
ISBN 0-8186-0755-6

THE COMPUTER SOCIETY OF THE IEEE

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

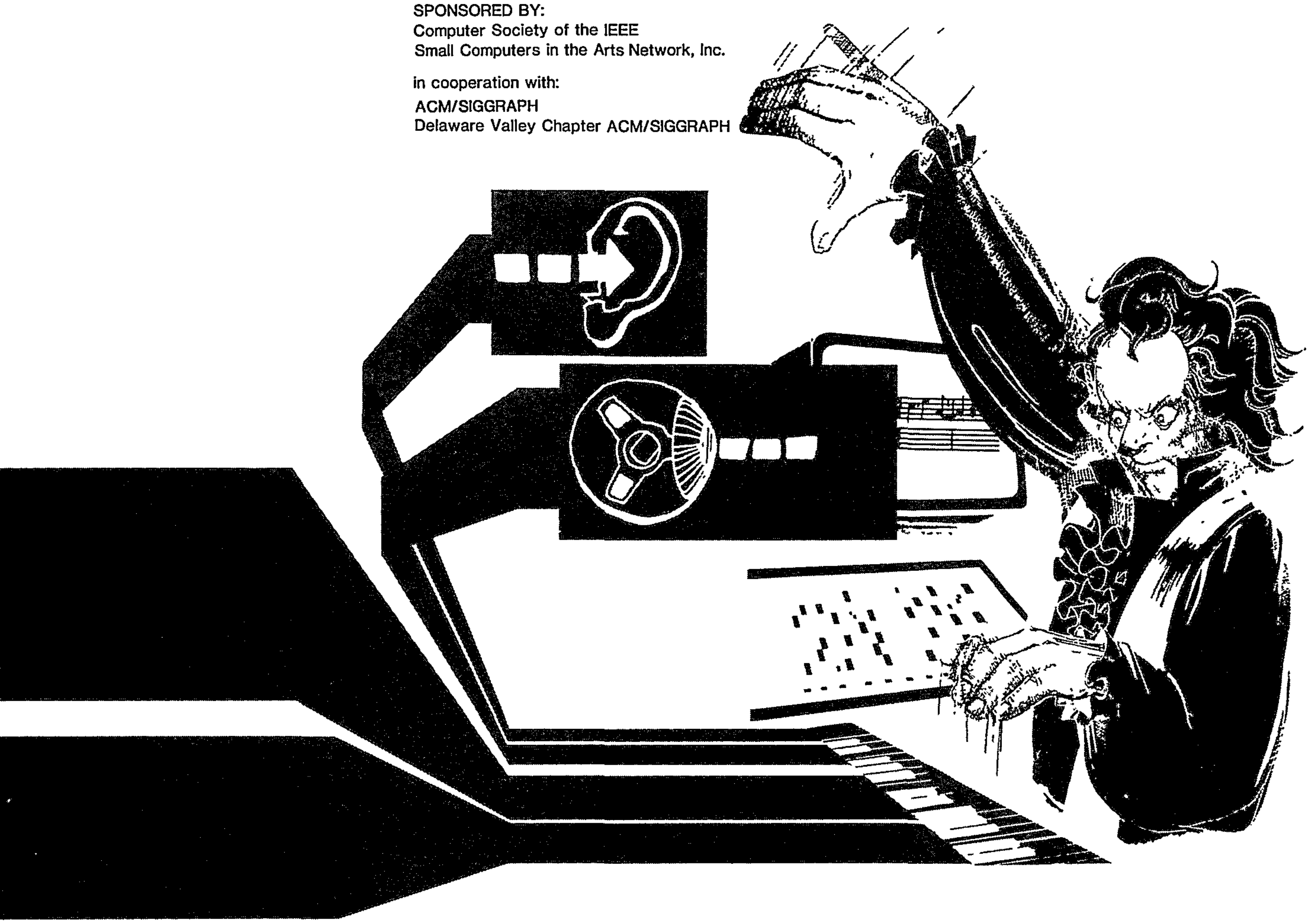
IEEE COMPUTER SOCIETY PRESS

PROCEEDINGS
6th symposium on small computers in the arts

OCTOBER 10-12, 1986
PHILADELPHIA, PENNSYLVANIA

SPONSORED BY:
Computer Society of the IEEE
Small Computers in the Arts Network, Inc.

in cooperation with:
ACM/SIGGRAPH
Delaware Valley Chapter ACM/SIGGRAPH



IEEE Computer Society Order Number 755
Library of Congress Number 86-45867
IEEE Catalog Number 86TH0164-4
ISBN 0-8186-0755-6

THE COMPUTER SOCIETY OF THE IEEE

IEEE
THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

IEEE COMPUTER SOCIETY PRESS

Published by IEEE Computer Society Press
1730 Massachusetts Avenue, N.W.
Washington, D.C. 20036-1903

IEEE Computer Society Order Number 755
Library of Congress Number 86-45867
IEEE Catalog Number 86TH0164-4
ISBN 0-8186-0755-6 (paper)
ISBN 0-8186-4755-8 (microfiche)
ISBN 0-8186-8755-X (case)

Order from:
IEEE Computer Society
Post Office Box 80452
Worldway Postal Center
Los Angeles, CA 90080

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

IEEE

1986 Proceedings of the Sixth Annual Symposium on Small Computers in the Arts

October 10-12, 1986
Philadelphia

Sponsored by:
IEEE Computer Society
Small Computers in the Arts Network, Inc.
and in cooperation with:
ACM/SIGGRAPH
Delaware Valley Chapter ACM/SIGGRAPH

Organized and Produced by:
Small Computers in the Arts Network, Inc.

INTRODUCTION

We would like to introduce these Proceedings of the Sixth Annual Symposium on Small Computers in the Arts by relating just a bit of the history of the group that organizes it, the Small Computers in the Arts Network, Inc. The group is committed to the promotion of small computers in the arts through a newsletter, concerts, the Symposium and various informal meetings.

This current group emerged out of an earlier one, the Personal Computer Arts Group, which held its first computer arts activity, a computer music concert, in 1978. The group hoped to provide a forum for creative people interested in using computers in the arts.

As the use of computers has grown in the arts, the Small Computers in the Arts Network has also changed and grown to reflect the greater penetration of small computers in the creative arts. We have just incorporated as a non-profit organization and have recently changed the format and expanded the focus of our newsletter, SCAN.

Much as our organization has grown and changed as computers have moved into a greater range of the creative and commercial arts, so have our proceedings. This year we have papers on computers in: animation, desktop publishing, sculpture, a data-flow language for music, and interactive storytelling.

The range of these papers demonstrates to us, the organizers of the Symposium, that a forum where computer artists can share knowledge and interests with their colleagues is as worthy a project today as we thought it was when we began six years ago.

Symposium Organizing Committee

Dick Moberg, Chairman
Alan Datri
Cathy DelTito
Trip Denton
Nancy Kimmons
Perry Leopold
Donna Mansfield
Bill Mauchly
Del McLean
Rob Morris
Rajen Naidoo
Tom Porett
Tom Rudolph
Mark Scott
Anne Seidman
John Senior
Joe Showalter
Maurice Wright

Small Computers in the Arts Network, Inc.
Box 1954, Philadelphia, PA 19105

TABLE OF CONTENTS

PATHWAYS TO ART THROUGH NUMBERS

MICHAEL F. DAUGHERTY

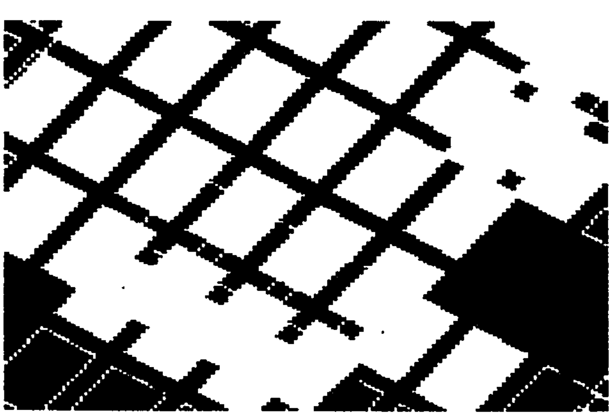
SCHOOL OF ART
LOUISIANA STATE UNIVERSITY
BATON ROUGE, LOUISIANA 70803

ABSTRACT

The foundations of art and design are intrinsically linked to the application of numbers and mathematics. Developed ,in part, under a grant from Apple Computer, Inc., Educational Affairs, the project "Pathways To Art Through Numbers" endeavors to integrate numerical content into a program of basic design education. The techniques and processes of computer generated imaging are utilized in design applications. The goal is to produce a practical design curriculum that combines computers, natural number systems and foundation design principles.

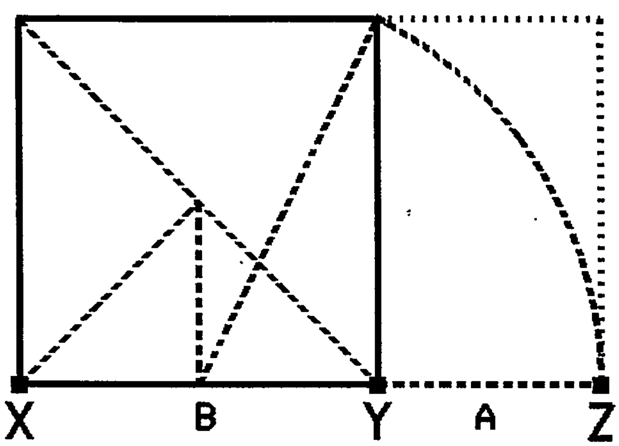
INTRODUCTION

The history of design is often viewed as man's mediation between the natural and the technological. Our latest tool, the computer, has the potential to heighten and expand this meditative process. Artists and designers can use the computer to access and assimilate numerical functions within fundamental design activities.



Composition based on golden section numbers, angles and rectangles.

The project, currently in its second year, blends numerical operations and design components in the development of solutions to traditional two-dimensional investigations. It combines research with useful classroom practices in such courses as basic design, graphic design and art structure.



GOLDEN RECTANGLE

$XY = B$

$YZ = A$

Φ = GOLDEN SECTION

$$\Phi = \frac{A}{B} = \frac{B}{A+B} = 1.618$$

"De Divina Proportione" Certain numerical systems, such as the golden section and golden rectangle, appear to manifest harmonic relationships. Mean / extreme ratios can be found in many natural structures and have been used by artists and designers for centuries.

TH0164-4/86/0000/0001$01.00©1986 IEEE

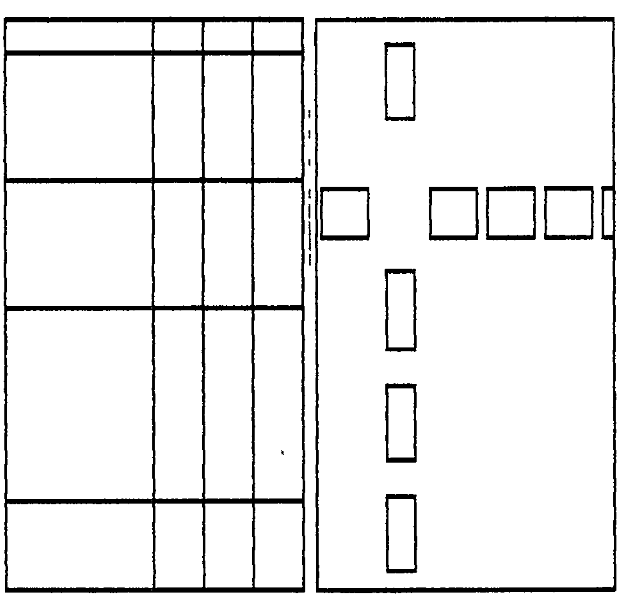
Elements of design (point, line, shape, etc.) are defined as unique computer procedures. Both internal and external variables are used to describe the elements. These are programmatically combined and manipulated by graphic structures (symmetry / asymmetry, figure / ground, repetition, transformation, patterning, etc.). The structures and elements are further qualified by recursion and value alteration (iteration, Fibonacci sequence, branching, random numbers, golden section, root-two, fractalization, etc.).

Design based programs are then written to experiment with the numerical calculations and test their usefulness on proportional and compositional relationships.

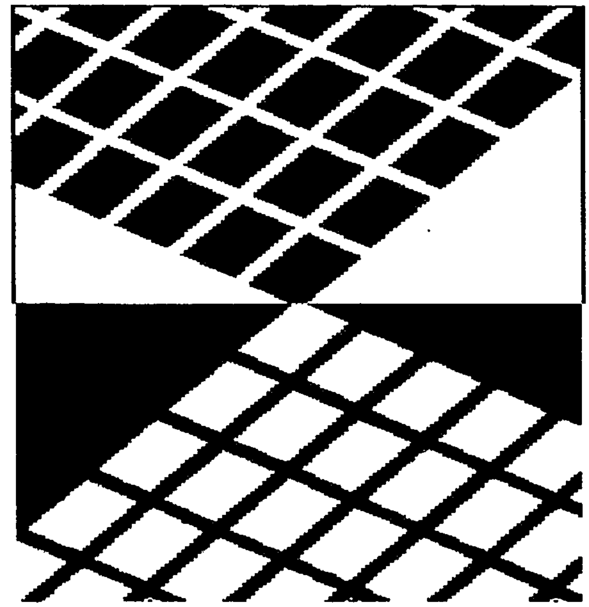
The following sections present some of the essential concepts and techniques used in the project. The images focus on the influence of golden section numbers. The golden rectangle uses the golden section number as an internal value to adjust the side / length ratios.

DESIGN OBJECTS

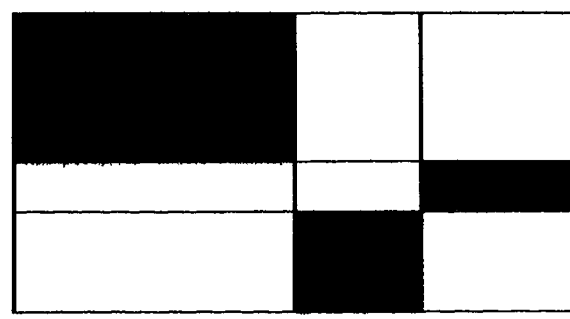
Design elements are drawn in linear form and defined as simple graphic objects. An object can be used as a single form or as a spacial modifier. Options such as perimeter fills and incremental size notation may be included.



Regulating lines and spaces were made with negative forms. Outlines of golden rectangles can be used to develop visible and implied structures.



A golden rectangular grid was used to study figure / ground and positive / negative relationships. Both grid and form spacing can be based on internal ratios.



Proportioning devices were developed for formal compositions. Positive and negative forms can be selected intuitively or by golden section divisional modifiers.

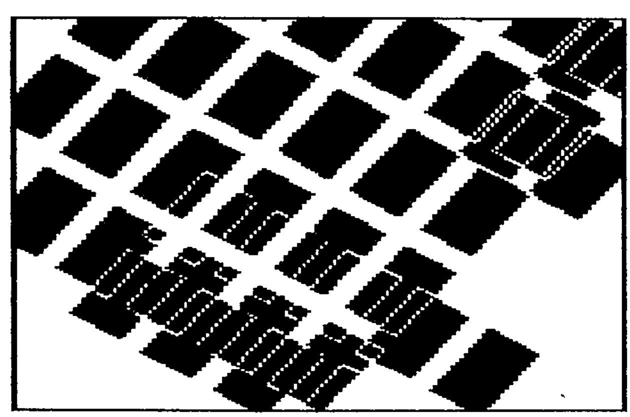
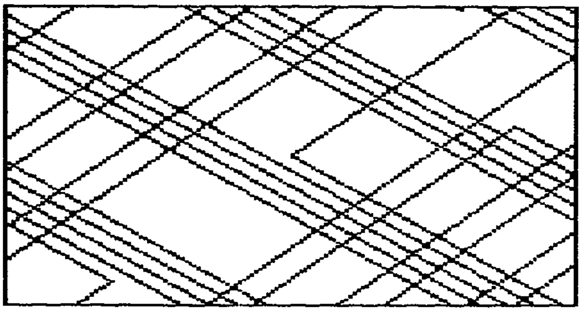
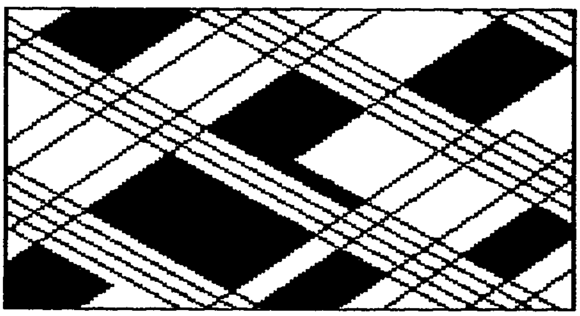


Image used superimposition of repeating postive and negative golden rectangles. Combined asymmetry with an implied rectilinear structure.





The golden section can be used to produce grids of various symmetries. In the process of creating an asymmetrically balanced composition, the golden section was used as a "distance selector" for positive area fills.

The definitions are written as procedures and use only relative coordinates. This format allows for flexibility and a more natural command of the objects. In contrast to methods that use set parameters this type of object description permits the user wide graphic narration.

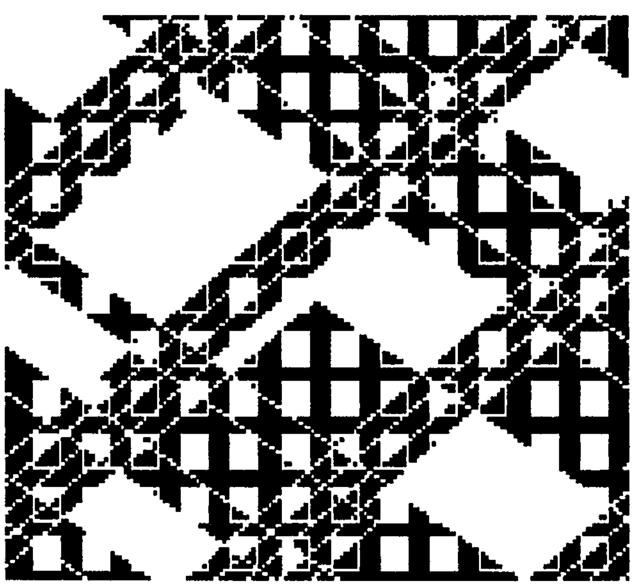
DESIGN OPERATIONS

Transformations are also obtained through programmed definitions. They are written with local variables and relative coordinates but are more general than object descriptions. Operations include both relative and absolute translations and rotations. Scaling operations are usually imbeded in the object definitions and accessed through user input or recursion.

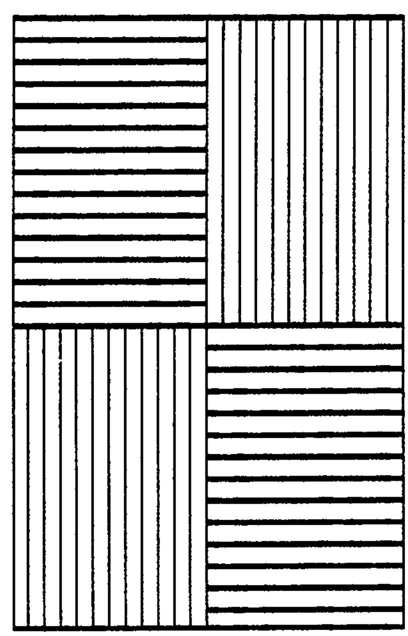
Values linked to variables may be preset but are customarily passed from the objects to the operations as incremental modifiers.

DESIGN INVESTIGATIONS

Ideally designers would want the most open, the least restrictive interactive environment. Providing a set of suitable design primitives is not especially difficult. But overly numerous, overly complex variable manipulations can cause problems and obscure the design process.



Composition used overlapping golden section grids. Area fills were controlled by "regulating" lines. This process created partial fills and a textural effect.



Uniform repetition of vertical and horizontal lines used the golden section as an area divisor.

The division between clarity and mud is often quite subtle; it seems there are as many possibilities as liabilities in writing effective image generating programs.

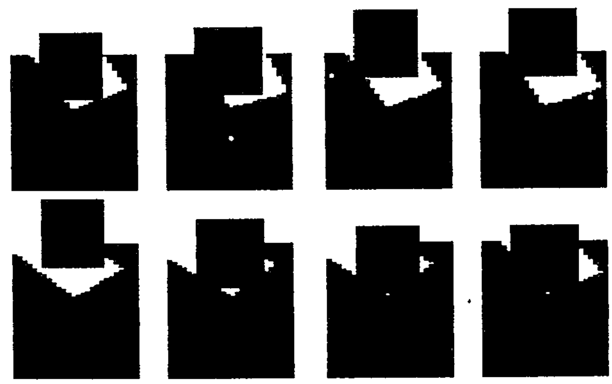
Certain limitations must be considered and built into the programs. Some of these have to do with specific machine and language requirements: screen resolution, addressable space, recursive functions, etc. Others relate to traditional tenets of pictorial composition. Programs should be written that foster creative exploration while testing for factors that might limit unexpected combinations or permutations.

RESEARCH GOALS

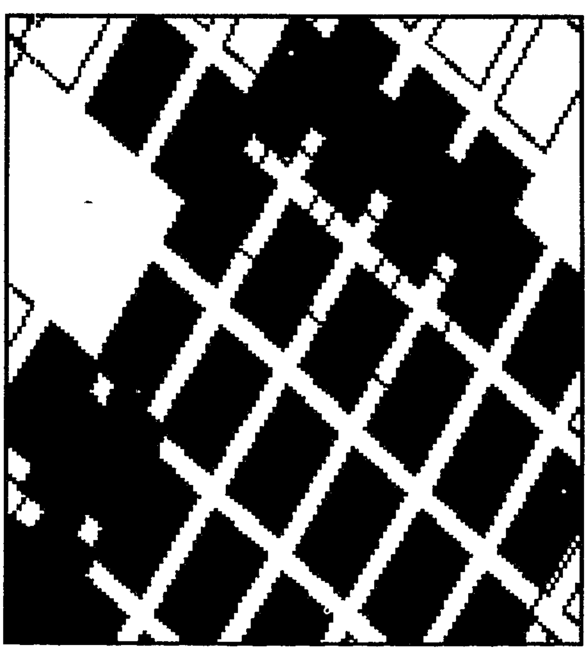
The entire field of design education is encountering a dramatic evolution. Computer graphics with its associated tools and technologies dominate the changes and challenges of this evolution. The development of software that aids in the communication of graphic ideas, concepts and information is the goal of the School of Art. The Pathways project aims to create an informational model that increases art and design inventiveness.

The School uses the College of Design's computer graphics lab for all of its computer based teaching and research.

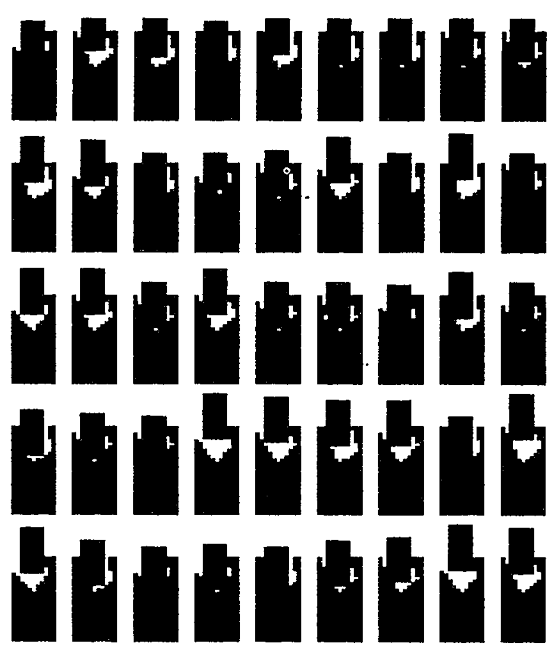
One of the projects now in the research and development stage is the "Design Mentor" system. This would be a unique visual language facility, enabling designers and artists to ask "what if" questions to many graphic problems. The construction of this utility would benefit the various disciplines within the College: architecture, art and landscape architecture. One aim is to reduce certain restrictive design distinctions at the introductory course level.



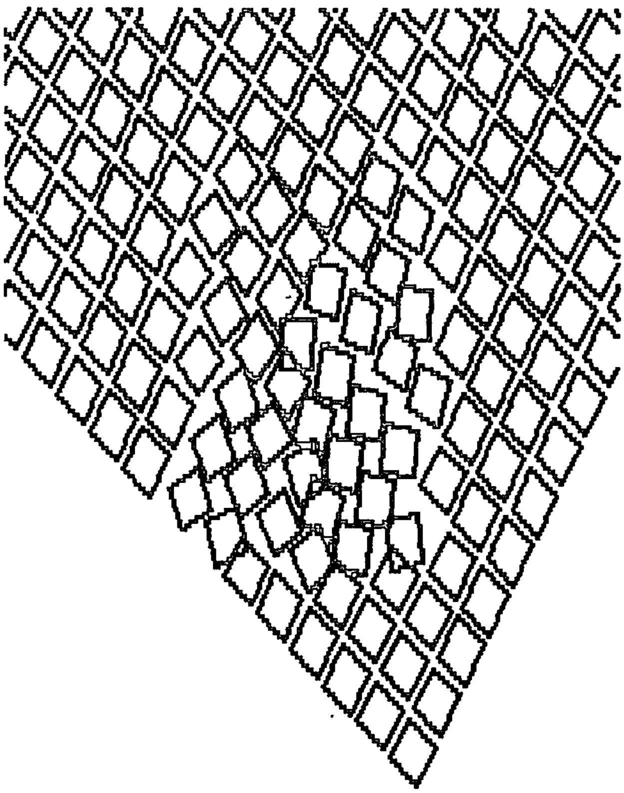
The combination of similar forms to generate complex units was aided by the application of golden section numbers. Forms were subtracted and added through a recursive process. The process also used random numbers.



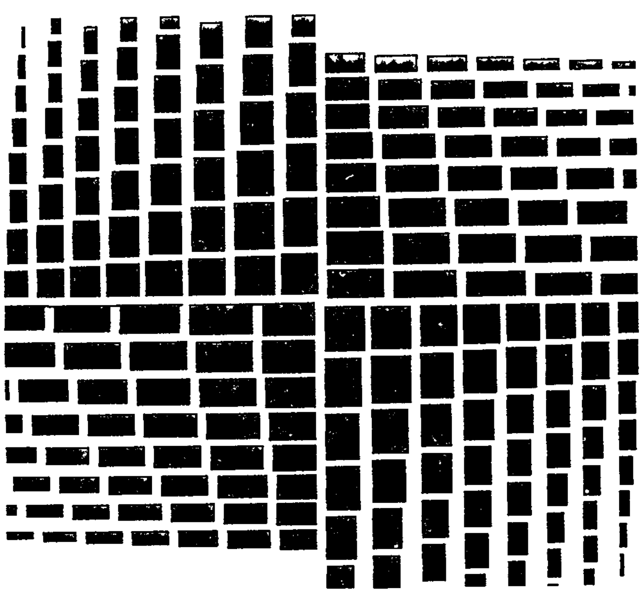
Reflection, scaling and mirroring of a golden rectangle. Reciprocals of golden section angles were used to alter the composition.



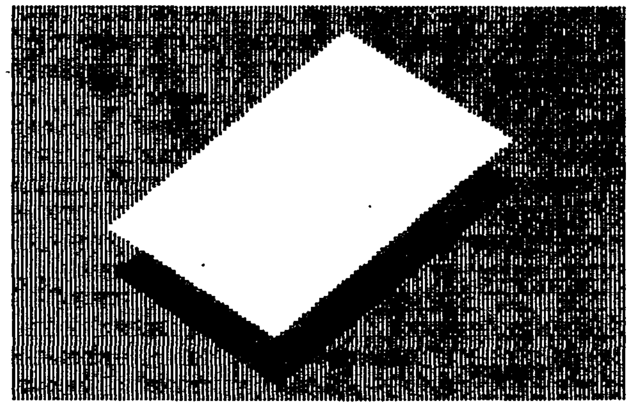
Similar unit and non-repeating patterns can be made by modifying the overall format with a "limited value" random number generator.



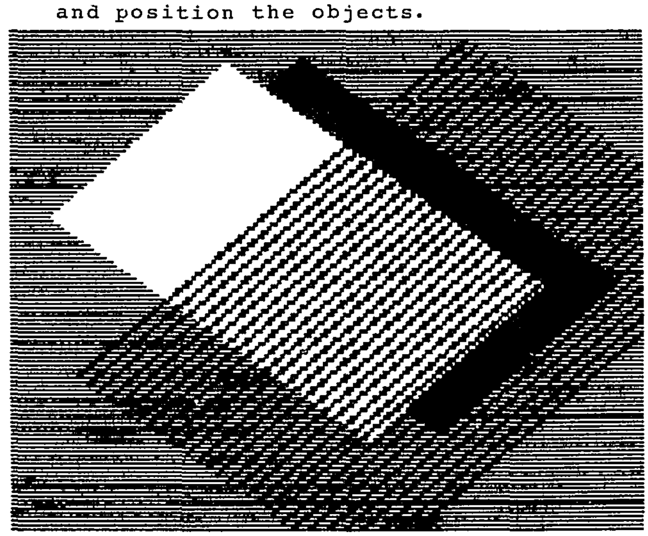
A pattern of squares was partially restructured through the use of random numbers. A predefined golden rectangular area was used to create pattern anomalies. If the current drawing point of a square entered the area a random angle number was called.



Recursion can be used to make progressive forms. Pattern gradation in four quadrants yields a radiating symmetry and a suggestion of receding forms.



Overlapping rectangles can imply depth. The image used the golden section to formally divide the space and position the objects.



An implied golden rectangle was created through linear repetition. Internal variables were employed to structure the entire image from one input: area. This type of program is being tested for possible electronic publishing applications.

CONCLUSION

The Pathways project is now moving from the discovery stage to the synthesis stage. Many numerical structures (natural, historic, anatomical, religious, political, etc.) are being evaluated for graphic linkage. Through study, testing and classroom activities the project hopes to make the computer imaging environment an even richer place for design research.

Every part of every object, shape, form and design is bound to mathematical rules. The power to express design concepts, influenced by numerical information is a challenging and exciting problem and one singularly suited to the computer.

COMPUTER DRAWINGS

Paul Rutkovsky

Assistant Professor, Department of Art, Florida State University
Tallahassee, Florida 32306
Fellow, Center for Advanced Visual Studies, MIT
Cambridge, MA 02139

I've always been curious about the similarity in the "look" of computer graphics, as most of us have been, whether they are super computer or just micro-computer images. I believe the major problem has simply been that scientists and engineers have had access to the equipment and that the population (some visual artists) at large is cyberphobic. It's easy to understand. Have you ever listened to a scientist explain simple shapes, such as a circle, square, or triangle in layman's terms, that is, in artists' terms? I have, and it kept me away from a computer for years.

I will briefly demonstrate, in simple language, how I've struggled with the limited two-dimensional rectangular picture plane of the MacIntosh and created a more flexible three-dimensional image.

Small computers are capable of creating fairly sophisticated three-dimensional images within the picture plane. However, all images are still limited by the very small visual working space on the terminal. Most so-called computer artists stay inside of the machine and do all of the struggling there. I believe that is a big limiting

Fig. 1 Digitized Portrait (detail), Paul Rutkovsky
38" X 55", 1986

TH0164-4/86/0000/0007$01.00©1986 IEEE

mistake and leads to the look-alike graphics anemia in the field. Too many visual artists print the final image and leave it at that. For me, that is only the beginning.

After the print is out,I put it either on the floor or wall and begin to add and subtract other images or use the Thunderscan or MacVision digitizers to "reconstitute" the image. This is where I utilize the full potential of the computer---outside the computer.

Much of what I do is simple collage, except when I keep redigitizing the same slightly altered image. After cutting out, pasting up, and digitizing several times, the transition from the original relatively rigid computer drawing to the final image is a lesson in freedom. The video camera and Thunderscan become another tool just like any special effects available for the MacIntosh.

Sometimes I believe that all of us who work with computers think there is some sort of unwritten holy rule that most computer activity should take place inside the machine. One large computer image I've recently been working on is

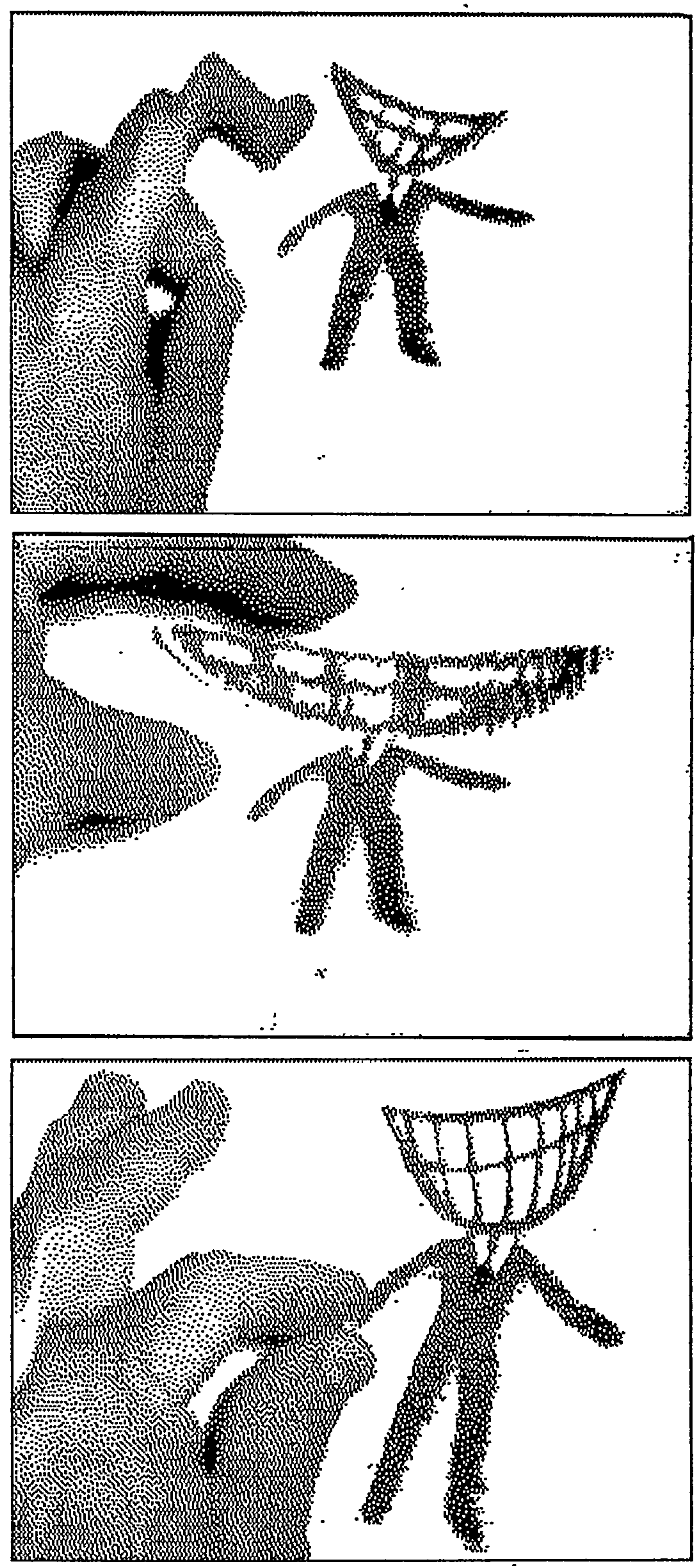


Fig. 2 Digitized Drawings, Paul Rutkovsky, 12" X 5" 1986 Each character was redigitized several times.

Fig. 3 Digitized Drawing (detail), Paul Rutkovsky, 40" X 82", 1986

40" X 82" and represents 20 to 30 hours on the digitizer (reconstituting) and another 10 inside the computer. I also modified textures and the beer cans and photo copied hundreds of these images. No need to wear the printer out.

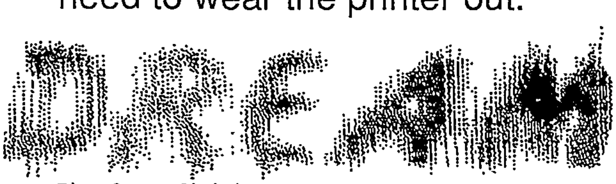


Fig. 3a Redigitized Type, Paul Rutkovsky, 1986

I did not start with computers. I worked with paint, clay, and photographs before I discovered computers. I consider myself to be a generalist and work with any media that obtains the results I'm looking for. I believe I've found a toy that will grow with time and become more flexible (fun and educational) in the not so distant future, especially if we learn to take it out of the machine and start playing.

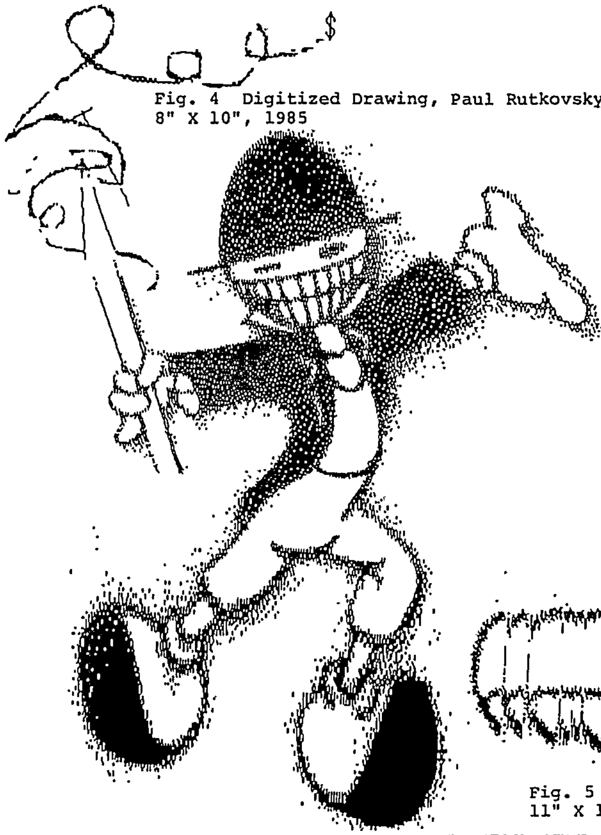


Fig. 4 Digitized Drawing, Paul Rutkovsky
8" X 10", 1985

Fig. 5 Redigitized Drawing, Paul Rutkovsky
11" X 14", 1986

Fig. 6 Redigitized Drawing, Paul Rutkovsky, 8" X 10", 1986

Algorithmic Art and Process Art

by
W.J. Kolomyjec PhD, MFA
Associate Professor of Art
School of Art
Northern Illinois University
DeKalb, Illinois 60115

Abstract

Algorithms were the earliest method for producing computer graphics / computer art. In this article the author defines algorithmic art and process art and gives examples. Process art is computer aesthetic imagery in the course of being done. Process art represents a new involvement with computer imagery and is facilitated by small machines.

Introduction

In the beginning there was no application software. If a person wanted to be creative with a computer or to be a computer artist s/he had to either write algorithms or collaborate with someone who could. Although there are a few books and manuscripts that predate Artist and Computer[1] edited by Ruth Leavitt, published in 1976, this publication is fairly descriptive of what computer art was like ten years ago. If one were to re-examine Artist and Computer a fairly significant observation can be made, namely, the majority of imagery presented in that volume is algorithmic, i.e., created by programming. Thus, it can be stated somewhat empirically that, at least ten years ago, the majority of computer graphics / computer art was created by algorithm. Incidently, this author has the distinction of being one of the contributors to Artist and Computer.

Webster's New World Dictionary[2] defines algorithm to be a mathematical term meaning "any special method of solving a certain kind of problem." A specific example would be Euclid's algorithm for finding the greatest common divisor of two numbers. In the context of computer graphics / computer art, algorithms are employed to create visual output. Algorithmic imagery is often distinctive, having a certain kind of quality or producing a certain kind of effect. Computer graphic algorithms can be as simple as a single routine for drawing regular polygons in vector or as complex as a complete software package (a collection of many algorithms) for generating and rendering three dimensional scenes using faceted, Gouraud or Phong shading, anti-aliasing and possibly ray tracing, in raster. In this article the term algorithmic art refers to that distinctive category of aesthetic computer imagery generated by some special method or algorithm. We will draw the line by saying it does not include unprocessed captured or grabbed imagery; imagery produced directly by an input device such as a scanner or camera digitizer. However, algorithm art could be generalized to include captured or grabbed imagery processed by algorithm (image processing.)

In this article this author would like to address the notion of algorithmic art from a personal perspective, therefore, I will shift to the first person I will illustrate algorithmic art with two example programs. However, the main intent of this article is to discuss algorithmic art in conjunction with small machines and how, with a microcomputer, process is as interesting as product or artifact. Process art is simply computer imagery in the course of being done, and I will illustrate this concept with an additional example.

Static Imagery

I am fascinated with plastic design. This goes back to my design school years when I was profoundly affected by Gyorgy Kepes' book entitled: language of vision. In this text, published in 1944, Kepes explained the term "plastic" as follows:

> Independent of what everyone "sees," every experiencing of a visual image is a forming; a dynamic process of integration, a "plastic" experience. The word plastic therefore is here used to designate the formative quality, the shaping of sensory impulses into unified organic wholes.[3]

As I look back, I realized that my "style" is not only consciously plastic but algorithmic as well. When I seriously began to make computer aesthetic imagery the only "application software" available was Ken Knowlton's EXPLOR. Although I liked what was possible with EXPLOR, I wasn't real thrilled with printer graphics so I forced myself to learn programming. In my early days I focused on programming the plotter. I often started with a little scribble or a graphical notion and generalized it into an algorithm for the computer to render into a visual statement.

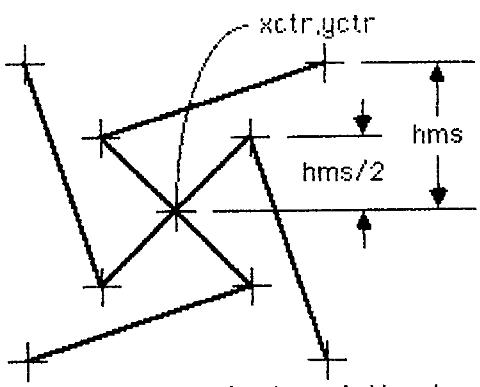


Figure 1. Module design for tesselation 1.

<u>Tesselations and randomness</u>

One of the best examples of algoritmic art that can be done with the computer is a tesselation. The simplest tesselation is a mosiac of squares which results in a grid. In a more general sense, a tesselation is a tiled surface where the tile(s) can be any shape as long as they fit together. Probably the greatest aesthetic tesselations were done by the famous Dutch artist M.C. Escher.

Tesselations can be easily programmed by computer by creating a generalized modular design and using a nested loop construct to calculate translation coordinates. Thus, an array of the module can drawn to form a tiled surface. The clever person will take care to design his/her module to interact with its neighbors, thus facilitating an overall symbiotic quality to the tesselation. Figure 1 represents a modular design and Figure 2 represents its tesselation. In the Appendix, Listing 1 gives the algorithm written in the C language for the Macintosh™ for those who are interested in programming.

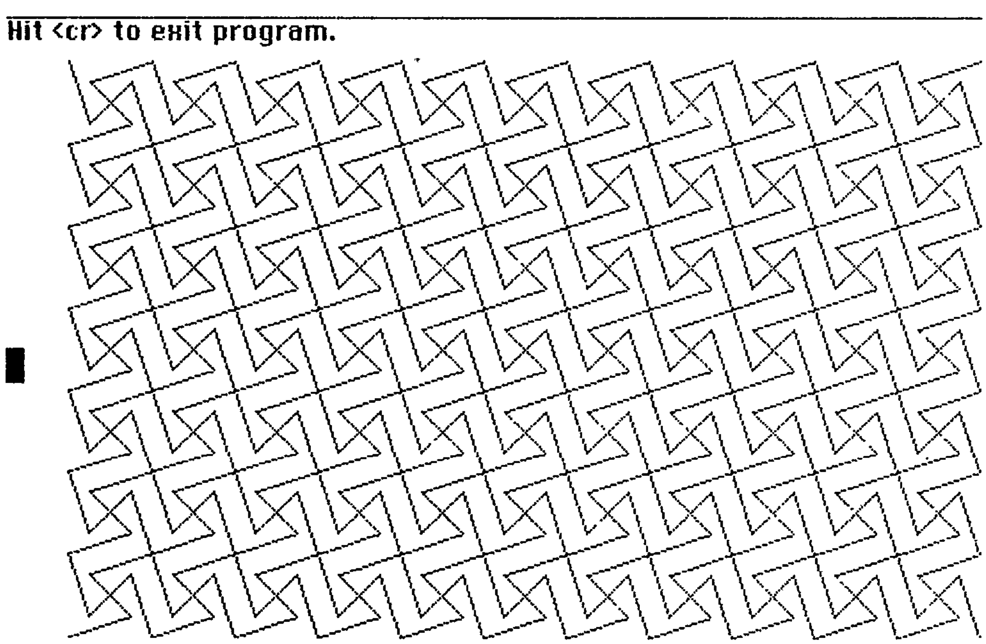


Figure 2. Tesselation 1.

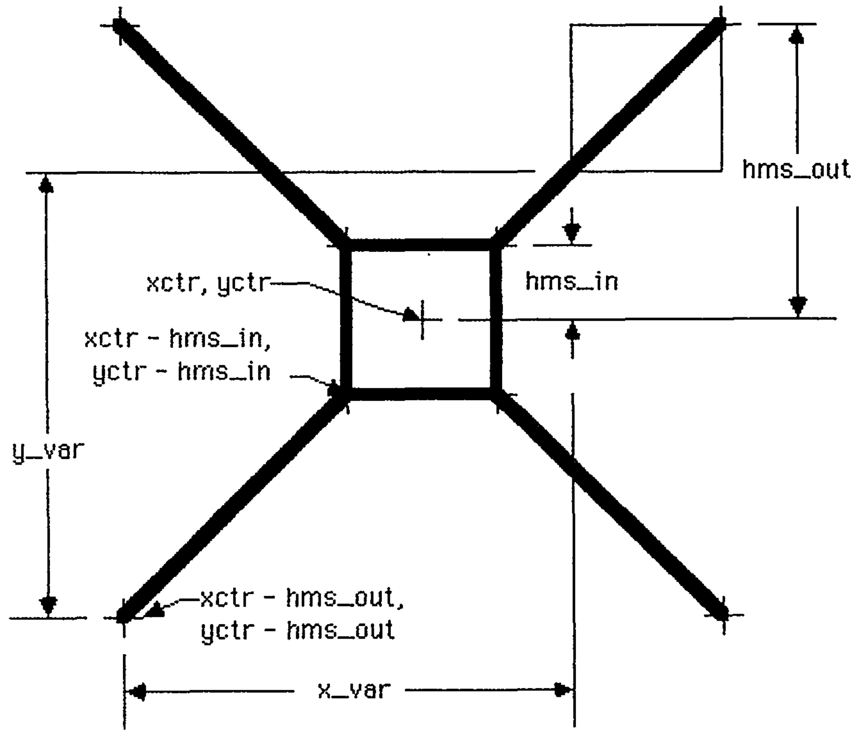


Figure 3. Design Module for Tesselation 2.

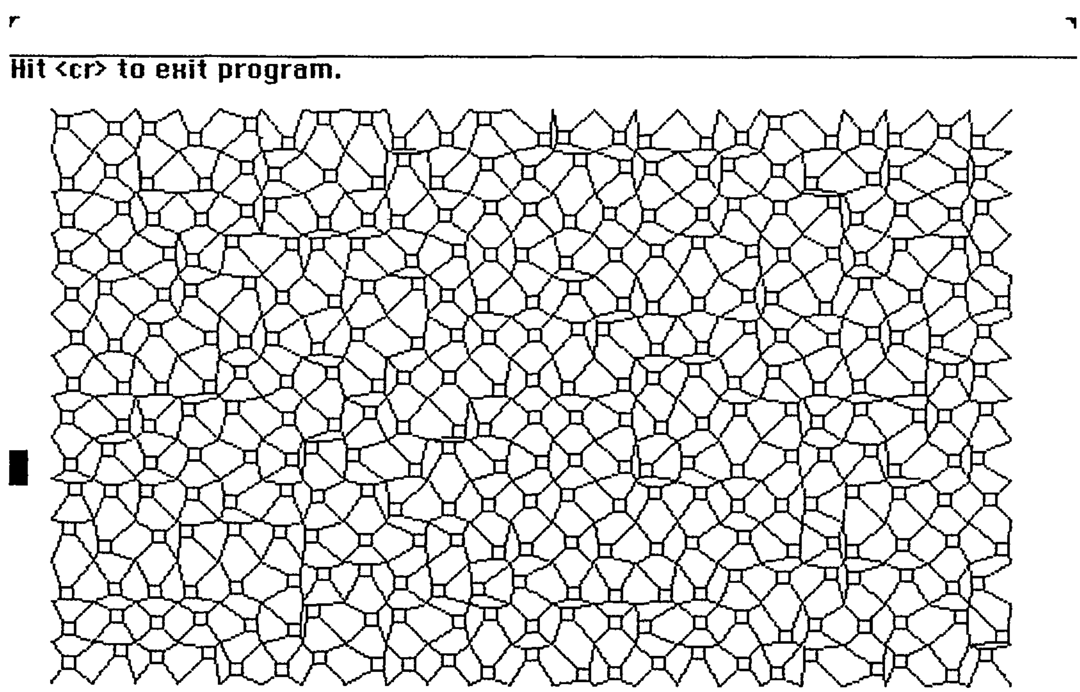


Figure 4. Tesselation 2 (The Web).

Randomness is another feature that can be incorporated into algorithmic art to add texture and variety. Almost every computer programming language has a built-in pseudo-random number primitive. Ranges of randomness can be specified by the algorithmic artist to perform a number of mundane or trival decisions. For example, quantities representing number, choice, translation, scale, rotation, etc., can all be parametized and incorporated into algorithms. In the Figure 3 another module design is given. It is a square (drawn) within a square (not drawn) in which the corners are connected with straight lines. In this example randomness is incorporated into the design to allow the inner square to vary left and right in x_var and up and down in y_var (see Figure 3.) Figure 4 shows the tesselation, I call it "the Web." In the Appendix, Listing 2 gives the algorithm written in the C language for the Macintosh™.

Animation

Many computer animated images have been produced by algorithm. A process called Keyframe animation is often used whereby two figures are provided to the algorithm and one is transformed into the other over a specified number of frames. The underlying algorithm in this process is called interpolation. Other algorithms can be used to control motion. Motion can be constant or uniform or images can appear to accelerate or decelerate during a transformation. I would refer the reader to another paper I authored entitled: "Keyframe Animation for the Microcomputer"[4] for a more detailed discussion of animation algorithms.

Process vs. Product

Several years ago I was being interviewed by a reporter for Computerworld Magazine[5] and I was asked what it was like to do computer art. I made the anaolgy that loading the program into the machine (a deck of cards into a card reader at the time) was like loading a shell into the breach of shotgun. Furthermore, executing the program was like pulling the trigger. At that point it was out of my hands and all I could do was sit back and watch. I also reflected that the procedure reminded me of 'action painting' as done by Jackson Pollack and other abstract expressionists. I had an image in my head which I tried to program the best I could, but when I compiled and executed it on the computer it was out of my hands. I reflected, I had in fact succeed in removing myself from the process of making my imagery; a goal sought by the so-called abstract expressionists. The humor in all this was that the interview was published with the title: "Concept behind 'shotgun' art approaches 1940s abstract expressionism."

This is an interesting phenomena related to computer art. The machine performs an artwork after-the-fact by following programmed algorithms; it becomes all process and in effect "time art." I used to sit for hours watching the plotter draw and this was as much a part of making the art as the final product. In retrospect I can make the observation that the product or the artifact is like the residue of my involvement with the image. My relationship with the artwork really begin with its programming and watching the piece execute was another level of interaction, this is what I call process art.

By way of example I submit Figure 5, a frame from an algorithmic / process art piece entitled "Scroller Balls." It is designed to run forever on a Macintosh™. A line of bit-mapped balls (filled ovals outlined in black) are rendered on the bottom of the display and then scrolled upward such that the next row may be rendered. This program, and the two given above, utilize the Macintosh QuickDraw ROM routines, available through Aztec C™. (These routines are also available through BASIC, but I just haven't tried using this interpreted language to do process art.) In the Appendix, Listing 3 gives the algorithm written in the C language for the Macintosh™.

Conclusion

In the very beginning all there was, was batch mode graphics; one was completely removed from his/her work and left only with a product. Today, using a microcomputer represents the antithesis of batch mode graphics. Microcomputers are machines that you can take into a studio or place on display in an installation. The newer machines have sophisticated and quick rendering built-in graphics. If you so desire, you can get personally involved in the image-making process. For example, by putting an infinite loop around a graphic algorithm you can produce a kind of perpetual image. Thus, product and process become integrated. This is a new and exciting dimension in aesthetic imagery facilitated by small machines.

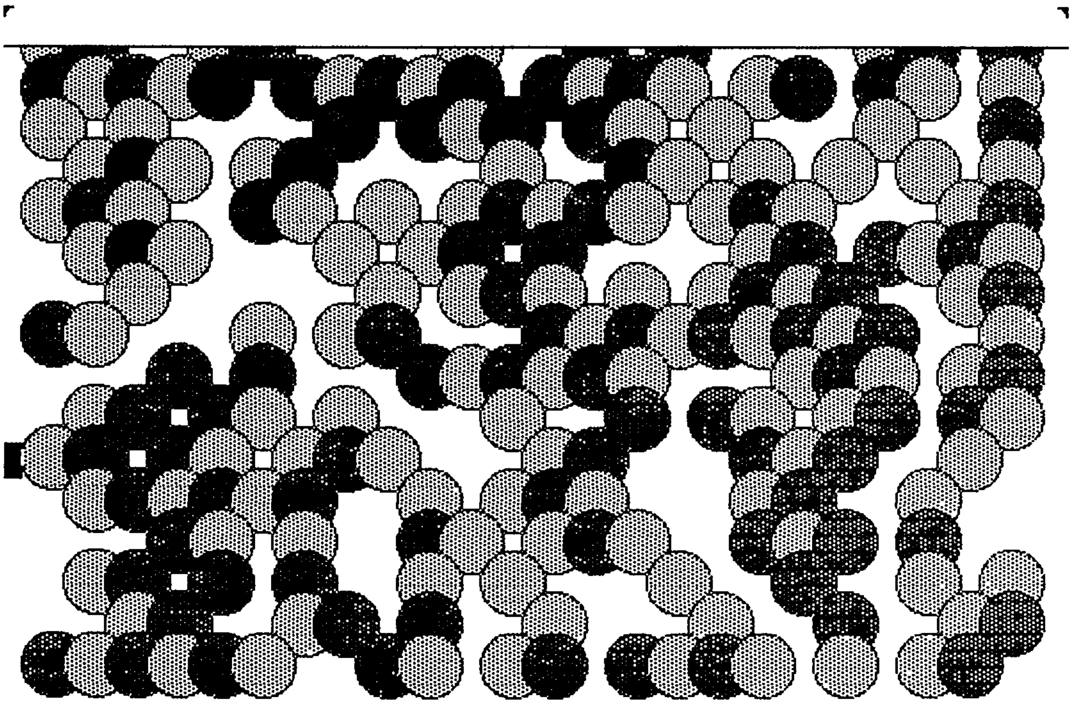


Figure 5. A frame from Scroller Balls.

References

1. Leavitt, R. ed. Artist and Computer. Harmony Books. New York, New York. 1976.

2. Webster's New World Dictionary of the American Language. Second Ed. Prentice-Hall, Inc. Englewood Cliffs, N.J. 1972.

3. Kepes, G. language of vision. Paul Theobald and Company. Chicago, Illinois. 1944. (out of print).

4. Kolomyjec, W.J. "Keyframe Animation for the Microcomputer." Proceedings. 3rd Symposium on Small Computers in the Arts. IEEE Computer Society Press. Silver Spring, Maryland. 1983.

5. Zientara, M. "Concept behind 'shotgun' art approaches 1940s abstract expressionism." Computerworld Magazine. July 5, 1978. pp30-31.

Appendix

```
/*
 *  Listing 1, Tesselation 1
 *  Algorithmic Art for the Macintosh
 *  written in Aztec C version 1.06H
 *  by Bill Kolomyjec
 */

#include <stdio.h>
#include <quickdraw.h>

main ()  {

/*
 *  define program variables and initial values
 *  screen coordinate units are in pixels
 */
    int x_offset = 50;
    int y_offset = 40;
    int hms = 20;

    int nx = 11;    /* number of modules per row */
    int ny = 7; /* number of moudules per column */

    int j,k;
    int xctr, yctr;
    static char msg[] = "Hit <cr> to exit program. ";

/*
 *  QuickDraw data type
 */
    Rect sr;

    void module();

/*
 *  define screen rectangle and clear screen
 *  QuickDraw routines
 */
    SetRect(&sr, 0, 0, 512, 342);
    EraseRect(&sr);

/* loop for rows */
    for (j = 0 ; j < ny ; j++) {
        yctr = 2 * hms * j + y_offset;

/* loop for columns */
        for (k = 0 ; k < nx; k++) {
            xctr = 2 * hms * k + x_offset;

            module(hms, xctr, yctr);

        } /* end of columns for */

    } /* end of rows for */

/*
 *   move to top of screen
 *   print exit message and hold screen
 */
    MoveTo (0,10);
    DrawText(msg, 0, 25);
    getchar();

} /* end of main */
```

```
/*
 *   single module design
 */
void
module(hms, xctr, yctr)
    int hms, xctr, yctr;
    {
/*
 *   h is one-half the half module size
 */
    int h = hms/2;
/*
 *   a generalized set of moves and draws
 */
    MoveTo (xctr - hms, yctr - hms);
    LineTo (xctr - h, yctr + h);
    LineTo (xctr + h, yctr - h);
    LineTo (xctr + hms, yctr + hms);
    MoveTo (xctr - hms, yctr + hms);
    LineTo (xctr + h, yctr + h);
    LineTo (xctr - h, yctr - h);
    LineTo (xctr + hms, yctr - hms);
    return;

} /* end of function */
```

```
/*
 *  Listing 2, Tesselation 2, The Web
 *  Algorithmic Art for the Macintosh
 *  written in Aztec C version 1.06H
 *  by Bill Kolomyjec
 */

#include <stdio.h>
#include <quickdraw.h>
#include <event.h>

main ()  {

/*
 *   define program variables and initial values
 *   screen coordinate units are pixels
 */
    int x_offset = 30;
    int y_offset = 35;
    int hms_out = 10;
    int hms_in = 3;

    int nx = 23;    /* number of modules per row */
    int ny = 14;    /* number of modules per column */

    int j,k;
    int xctr, yctr;
    static char msg[] = "Hit <cr> to exit program. ";

    Rect sr;
    void module();

/*
 *   set random seed backed on tickcount
 *   randSeed is a QuickDraw routine
 *   TickCount is a Macintosh function
 */
    randSeed = TickCount();
```

```
/*
 *   define screen rectangle and clear screen
 */
    SetRect(&sr, 0, 0, 512, 342);
    EraseRect(&sr);

/* loop for rows */
    for (j = 0 ; j < ny ; j++) {
        yctr = 2 * hms_out * j + y_offset;

/* loop for columns */
        for (k = 0 ; k < nx; k++) {
            xctr = 2 * hms_out * k + x_offset;

            module(hms_out, hms_in, xctr, yctr);

        } /* end of columns for */

    } /* end of rows for */

/*
 *   move to top of screen
 *   print exit message and hold screen
 */
    MoveTo (0,10);
    DrawText(msg, 0, 25);
    getchar();

} /* end of main */

/*
 *  single module design
 */
void
module(hms_out, hms_in, xctr, yctr)
    int hms_out, hms_in, xctr, yctr;
    {

    int x_var, y_var;
    double rnd();

/*
 *   one line algorithms for random
 *   variance in x and y directions
 */
    x_var = (2.0 * rnd() - 1.0) * (hms_out - hms_in);
    y_var = (2.0 * rnd() - 1.0) * (hms_out - hms_in);


    MoveTo (xctr - hms_out, yctr - hms_out);
    LineTo (xctr - hms_in + x_var, yctr - hms_in + y_var);
    LineTo (xctr - hms_in + x_var, yctr + hms_in + y_var);

    MoveTo (xctr - hms_out, yctr + hms_out);
    LineTo (xctr - hms_in + x_var, yctr + hms_in + y_var);
    LineTo (xctr + hms_in + x_var, yctr + hms_in + y_var);

    MoveTo (xctr + hms_out, yctr + hms_out);
    LineTo (xctr + hms_in + x_var, yctr + hms_in + y_var);
    LineTo (xctr + hms_in + x_var, yctr - hms_in + y_var);

    MoveTo (xctr + hms_out, yctr - hms_out);
    LineTo (xctr + hms_in + x_var, yctr - hms_in + y_var);
    LineTo (xctr - hms_in + x_var, yctr - hms_in + y_var);
    return;

} /* end of module function */
```

```
/*
 *   random number function which
 *   returns a value between 0.0 and 0.9999...
 */
double
rnd() {

    return ((Random() & 32767) / 32768.0);

} /* end of random function */
```

```
/*
 *  Listing 3. Scroller Balls
 *  Algorithmic / Process Art for the Macintosh
 *  written in Aztec C version 1.06H
 *  by Bill Kolomyjec
 */

#include <stdio.h>
#include <quickdraw.h>
#include <event.h>

main ()  {

/*
 *   define program variables and initial values
 *   screen coordinate units are pixels
 */
    int xoff = 24;
    int radius = 16;
    int dist = 20;
    int xctr, yctr = 300;

    int j,k;
    int nx = 24;
    int flip = 0;

    void module();
    double rnd();

/*
 *   QuickDraw data types
 */
    Rect sr, da;
    RgnHandle rh;

    SetRect(&sr, 0, 0, 512, 342);
    SetRect(&da, 0, 0, 512, 320);

    rh = NewRgn();

/*
 *   clear screen
 */
    EraseRect(&sr);

/*
 *   initialize the random number generator
 */
    randSeed = TickCount();

/*
 *   infinite loop for rows
 */
    for (;;)
    {

/*
 *  draw a row of modules, throw out 50% of them
 */
        for (k = 0 ; k < nx; k++)
        {
            xctr = dist * k + xoff;

/*
 *   coin toss algorithm
 */
            if ( rnd() - 0.5 < 0.0 )
                module (xctr, yctr, radius, flip);
```

```
/*
 *   use NOT operator to flip
 */
                flip = !flip;

        } /* end of rows for */

/*
 *   extra flip because number in row is even
 */
        flip = !flip;

/*
 *   scroll drawing area up the distance of a row
 */
        for (j=0; j<dist; j++)
        {
            ScrollRect(&da,0,-1,rh);

/*
 *   a way to exit infinite loop
 *   exit if mouse button pushed
 */
            if (Button())
                exit (-1);

        }  /* end of scroll loop */

    } /*end of infinite loop */

} /* end of main */


/*
 *  module subroutine, draw a framed circle
 *  whose color (tone) is determined by flip
 */
void
module (xctr, yctr, rad, flip)
    int xctr, yctr, rad, flip;
    {

    Rect rm;
    int t, l, b, r;

    t = xctr - rad;
    l = yctr - rad;
    b = xctr + rad;
    r = yctr + rad;

    SetRect(&rm, t, l, b, r);

/*
 *   draw the ball ltGray or dkGray
 */
    if (flip)
        FillOval(&rm,dkGray);
    else
        FillOval(&rm,ltGray);

/*
 *   frame the ball in black
 */
    FrameOval(&rm);
    return;

} /* end of module subroutine */
```

```
/*
 *  random number function which
 *  returns a value between 0.0 and 0.9999...
 */
double
rnd() {

    return ((Random() & 32767) / 32768.0);

} /* end of random function */
```

TEACHING GRAPHIC DESIGN WITH A MICROCOMPUTER

Lauretta Jones

School of Visual Arts
New York City

Abstract

Responding to a gap I perceived in the computer instruction available to artists in the New York City area, I developed a course at the School of Visual Arts which introduces graphic designers to a microcomputer relevant to their needs: the Macintosh. I took a pragmatic approach, presenting the microcomputer as a design and production tool, while stressing the need to maintain hold of aesthetic values in grappling with the mastery of new ways of working. Teaching the course in both the degree and continuing education programs, I found the contrast of students to be both personally entertaining and also instrumental in the ongoing development of this course.

Introduction

In teaching the introductory and intermediate computer graphics workshop courses in the continuing education program at the School of Visual Arts in New York City, I found each class to hold a motley assortment of fine artists, commercial designers, illustrators and art directors, cartoonists, teachers, programmers, animators, thrill-seeking secretaries, the curious and the frightened.

In both workshop classes – one taught on Apple II+s, the other on IBMs – I sought to take advantage of the small class size to respond to the range of student interests. Class size ranged from eight to eighteen students due to SVA's one-student-per-computer policy.

After initial basic instruction, I focused my attention on each student's individual needs and interests – as far as software, my knowledge and time permitted.

Despite my years of experience in the graphic design field (or perhaps, due to that), I found there was not much I could recommend to my graphic design oriented students on either the Apple or the IBM. Therefore, when SVA received a shipment of ten 512K Macintoshes in early 1985, I was ready and waiting with the outline of a new course entitled, "Graphic Design on a Microcomputer."

"Don't throw out your tracing paper yet," I teased in the course description...

Computer of Choice

There are many things which recommended the Macintosh as the machine on which to base this course. The Mac's much-touted what-you-see-is-what-you-get, black-on-white, one-to-one-with-the-real-world-ratio screen display, its relative ease of use, the availability of good graphics peripherals, software and typefaces, the 300 dots per inch resolution of the LaserWriter, and the availability of Linotronic typesetting output all contributed to my decision and to its popularity among small studios and individual designers.

In the conceptual or beginning phases of the design process, the Macintosh has the speed and drawing aids to keep up with the often rapid flow of an artist's ideas. In the production end of the design process, the Macintosh has practical applications in preparing comps (comprehensive layouts), developing mechanical illustrations and producing formatted publications. Layout revisions can be accommodated electronically before the manuscript is typeset, eliminating the cost and time of type and pasteup corrections.

Although, strictly speaking, it is outside the curriculum of this particular course, the Macintosh can prove useful for the accounting, invoicing and mail list needs of designers. That is, if the secretaries can pry the artist's fingers from the mouse.

Soft and Hardware

The School of Visual Arts began with ten

TH0164-4/86/0000/0021$01.00

Macintosh 512K computers which we've recently replaced with twenty Enhanced Macs (still 512K of RAM, but with new and faster ROMs and 800K, double-sided internal disk drives). We are currently retaining the single-sided external drives. The students are introduced to ThunderScan and Magic video digitizers, the Summagraphics and GTCO graphics tablets, and the ImageWriter I and LaserWriter printers. They complete assignments using MacPaint (bit-mapped paint program), MacDraw (vector drawing program), MacWrite (word processor), PageMaker, ReadySetGo, and MacPublisher (all page layout programs), ClickArt Special Effects and Fonts, and various other fonts and desk accessories.

Each student, in addition to having sole access to a computer during class time, has ample opportunity to work in SVA's computer center between classes at no additional charge. ID cards are issued, three-hour slots booked, and the center keeps a seven day a week schedule.

Course Development

Another effect of the broad scope of the student's interests in my initial workshop classes was often too much time spent on technical instruction at the expense of aesthetic and design issues. In tailoring the assignments in this class to graphic designers, I was better able to integrate my instructional goals.

All that boring basic stuff... My powers of persuasion seem to grow with my experience in teaching this course, yet "Desktop" maneuvers are never a big hit with a new class. The Desktop is the first screen which greets you upon booting a Macintosh: it is a dynamic metaphor as well as a place where the contents of disks can be scrutinized, copied, renamed, and deleted. Here I begin the course, demonstrating some basic computer principles as well as the conceptual and functional aspects of the Macintosh graphic interface. Here is where students begin to develop their coordination in "mousing" techniques.

Many students scoot directly into MacPaint, whence I have to drag them back. It doesn't hit them until later how much time and ingenuity they could save by checking disk space, contents and names before they find themselves smack-dab in the middle of a "Save" attempt to a disk filled with unneeded files.

I begin the assignments with a logo in MacPaint. The fun quotient of this program serves to diffuse any lingering fears in students who are confronting a computer for the first time. Remember, my students are primarily artists who have for most of their lives held that mathematics and technology are not only irrelevant but detrimental to the way they work.

In developing thumbnail sketches in MacPaint, students find that creating rough designs on the computer can have certain advantages over roughs on tissue paper. That is, after they get over the inevitable strangeness of a totally new tool.

The Clipboard and Scrapbook are the software tools by which images and text can be transported between Macintosh programs. For anyone who has used another computer, this is a delightful advantage. However, despite their cheery names, I find their use to be my students' most consistant stumbling block. I offer the students an incentive in the form of a promise – once they convince me they know how to use the Clipboard and Scrapbook, I'll show them how they can avoid it.

I originally assigned an invoice in MacWrite. Since I had been maintaining my own invoices in this manner, it seemed like a good idea. But none of the students could be convinced to use the ImageWriter after the LaserWriter was set up, and we soon ran into so many problems with the LaserWriter's interpretation of rules, underscores, etc. that I reconsidered.

I introduced another program into the syllabus. The students now developed their invoice forms with superior results in MacDraw, and I shortened the MacWrite assignment to a quick in-class letter to Dad. In both assignments the students had to import their MacPaint-developed logo through the Clipboard and Scrapbook.

My primary concerns were that they learn the editing functions of MacWrite, the characteristics of a vector-based drawing program such as MacDraw, and how each of these programs (unlike MacPaint) afforded the highest quality of type possible on the LaserWriter.

In choosing ThunderScan to digitize images, I stressed the advantages of controlling the contrast and brightness of discrete areas of the image. This is especially helpful when converting a color image to black and white. The ThunderScan system includes a scanning head which replaces the ribbon cartridge in the ImageWriter printer, digitizing a paper-based image line-by-line. Whereas a digitizer using a video camera can capture an image in seconds or a fraction of a second, ThunderScan can take up to twenty minutes to scan an eight by ten photo or drawing. This time disadvantage is often mitigated at SVA by scanning with one computer while continuing drawing or typing on another. I have walked into the computer center to find a student working on three computers at once.

Scanning devices of all kinds present a temptation to casually violate copyrights. The headline for one digitizer actually reads "It Took DaVinci Six Years – It'll Take You 1/30th Second." While the Mona Lisa is public domain, and therefore fair game, this illustrates a pervading additude which tends to blur the issues of copyright and fair usage. Believing that knowledge of the copyright statutes is essential to the financial well-being of every artist, I take the opportunity to pass this information on to my classes. Even with the continuing education students, it is all too often the first time they have ever heard of their rights under the law.

This all leads up to PageMaker. Originally, I demonstrated MacPublisher and ReadySetGo as well and let the students choose the page layout program they felt comfortable with. Most used PageMaker. I decided to demonstrate a single page layout program several times rather than several programs once each. This enabled both me and the students to concentrate on good design as well as production techniques.

For this assignment, students were to design and produce a double-page spread and the cover of a fictitious publication. A magazine, annual report, catalog or brochure were all possibilities. Here was an opportunity for them to bring together and practise the skills learned in using MacPaint, MacDraw, MacWrite and ThunderScan. Roughs were completed in MacPaint; bodycopy in MacWrite; headlines in MacPaint, ThunderScan, MacDraw or PageMaker. If the students professed interest at this point, I described the technical aspects of vector, bitmapped and ASCII fonts in addition to the printed results of each. They created images in MacPaint, ThunderScan or MacDraw; with the final assembly, editing and "massaging" taking place in PageMaker.

Ending on an upnote: Magic. After the long (3-4 week) and involved page layout assignment, I threw in a quick one week "fun" assignment. Manning the lights, video camera and computer, I captured portraits of each student, following their art direction. I found the control of Magic software sufficiently unclear (with sparse and ill-organized documentation) to warrant this "hands-off" approach.

With four or five variations of their faces and body parts safely on disk, each student then had their choice of assignments using this image: the cover of their long-awaited autobiography, a metamorphosis of their image in five steps or more, a shopping bag design for Bloomingdales, a package design for simulation software which simulates themselves. This assignment was to be fun for me as well, as I stepped back and let the students go, refusing to comment or even look at rough designs. It was interesting to note which students shone through this abrupt change in interaction.

Deeper Responses to a New Tool

I suggest that my students investigate what I believe to be the strong points of the computer for artists. In the initial stage of designing – developing and refining basic concepts – the speed of the computer can often keep pace with the designer's flow of ideas. This stage is marked by a profusion of "what-if" variations on the initial rough sketches. What if the logo were reversed, realigned, moved around, enlarged, reduced, reproportioned, enclosed, combined with another rough idea? In my own work, I often found that while committing logo versions two and three to paper, versions four through eight might quickly rush into – and right back out out of – my mind. With the Macintosh, thumbnails proliferate into scores of variations. And what's more, a day later I still understand what the sketches indicate.

Fear of overworking a design and the reluctance to pursue any solution not "guaranteed" to be the "right" one ties up the creative efforts of fine and commercial artists, students and professionals alike. I encouraged students to stop and save their work when they feel themselves to be at a decision point. As they never need commit themselves to a brush or penstroke on an original final, they can take advantage of the redrawing time saved by the computer to pursue more of their creative options. I've been gratified to see some students breaking out of their safe, tried-and-true-but-boring design solutions when presented with this electronic "safety net."

Another effect of transferring the graphic design proccess to a computer is a breakup of its sequential nature. Ordinarily, the designer and production artist find themselves squeezed against a press deadline at the end of a process in which they've been preceded by an account executive, a writer, an editor; and out of which remaining time they must account for the turn-around needed for typesetting, photography or illustration, client approvals, and revisions. With the computer, the distinctions between a rough design and its final, camera-ready form are softened: roughs can literally metamorphose into comps and camera-ready art. Design and "mechanical" preparation need not wait for final copy approval, as changes can be made more cost effectively. If the writer prepares her work on a compatible word-processor, copy-fitting and type specing may become anachronistic skills. If the skills of a writer, illustrator, designer and production artist all reside in the hands of an individual, they now have the means to realize a finished piece.

Continuing Education and Degree Students: An Advantage in Contrasts

Continuing education students had spent an average of one to ten years in the field as designers, mechanical artists, typesetters or editorial art directors. They had established styles and ways of working and, although their designs were usually polished and professional-looking, they were not prone to experiment and take risks.

I often had to loosen some of these students from the "way they worked" – which they would desperately try to recreate on the computer. I would encourage them to leave their preconceptions behind and view the Mac as an adventure, as an experiment in discovering new techniques. Some, of course, never overcame their initial disappointment, and failed to find those things the computer had to offer. Others delighted in developing techniques and discovering hidden keyboard combinations which the teacher hadn't demonstrated and maybe didn't even know about!

Perhaps a third of the continuing ed students already owned a Macintosh or used one at the office and wanted to "learn what to do with it." Another third were looking to make a purchase decision or expecting their offices to acquire a computer. Still others were motivated by curiosity or a fear of becoming obsolete. Those coming out of fear often didn't complete the course.

The degree students, on the other hand, were very raw both in their design sense and knowledge of production techniques. After one experience with a freshman who didn't know what I meant by "thumbnail sketch," "font," or "pica," I requested my course be limited to upperclass students with some design credits behind them. I spent much time speaking to degree students about basic design issues, visual and verbal presentation techniques, and what it was like out there in the "real world."

I found that, as with the continuing ed classes, the focus of the course on graphic design rather than the more ambiguous "computer graphics" afforded me the opportunity to deal more directly and in more depth with good design. The degree students were more apt to respond to my comments and suggestions – sometimes a little too slavishly for my satisfaction – but this included a greater willingness to venture into uncharted territory.

I required all students to make a verbal presentation of their logo and page layout assignments. They were to talk about the thinking behind both their aesthetic and technical decisions. Many enjoyed this opportunity to commiserate about software that had bugs, computers that bombed, printers that took three hours to print a single page. The presentations were often punctuated with "ahh"s, and "how-did-you"s, as students took advantage of vicarious learning. Many also found it a great reassurance that they were not the only ones to see themselves in a pitched battle with a cute, recalcitrant hunk of silicon and plastic.

Although I personally do not see it as such a great advantage, many students liked how clean working with a computer could be: no ink or graphite stains on hands and forearms, no little bits of waxed paper stuck to strange places. Many appreciated a reduction in rubber cement, solvent and marker fumes. Some bemoaned the Macintosh's lack of color, yet most found this no handicap, as they were used to designing with pencil or black marker and adding color later. Production artists must reduce color to black and white anyhow.

The students offered good practical critiques of software, some of which I've passed on to the appropriate publishers. ReadySetGo lost to PageMaker on such points as an inability to layer text and images without their overlapping boxes cropping undesirably; MacPublisher's awkward handling of graphics via its "camera" and generally difficult interface resulted in student's shunning the advantages it offered in kerning capabilities.

Future Directions

The course I have outlined above is taught in a single semester – fifteen sessions for the degree school, twelve or ten for continuing education. I pack a lot into those sessions, and the students never really have an opportunity to catch their breath. They come out of the course with a necessarily shallow understanding of a broad range of soft- and hardware; it often seems the semester ends just as they reach a level of comfortable competance.

In the coming school year, the course will be taught in two semesters. I've yet to decide whether to maintain the pace of the first semester and use the second for portfolio development, or to introduce the programs more gradually over both semesters. Three other instructors are now teaching this course, and I look forward to comparing notes with them.

WHY ARTISTS SHOULD PROGRAM

Craig Hickman

University of Oregon
Department of Fine and Applied Arts

Abstract
As the computer is more accepted in the arts much, of what we now call computer art will be absorbed by existing media. At the same time, computer art will develop from a minor, derivative medium to one with its own distinctive (and as yet mostly unimagined) characteristics. New opportunities will abound for the artist to develop the possibilities that the computer presents. Programming is the most effective way to explore and develop these possibilities. It should be taught to artists in a way tailored to their visual orientation, perhaps using one of the new versions of computer languages which make visual programming less frustrating for the novice. Artists can also make a contribution to software development in general.

The Changing Role of the Computer in the Arts

As graphics program packages become more sophisticated and specialized, the resolution and quality of computer output media increase, and artists become more aware of the computer's potential, it is generally assumed that artists will no longer need to learn to program. After all, painters don't weave their canvases, photographers don't build their cameras, and poets don't construct their typewriters, so why should computer artists write their software from the ground up? It is my belief, however, that the most exciting and important applications for computers in the arts will come through artists programming. The reason lies fundamentally in the nature of the computer and the role of the artist.

I do not mean to say that all artists using computers should write their own software. In fact, most shouldn't. Much exceptionally good graphics software is now available, and it would be redundant to re-write it. But the definition of "computer graphics" is constantly changing, so that many software applications now considered as computer graphics will pass to their traditional disciplines. We graphics artists, for example, introduced Paint and Draw programs to our friends who paint and draw, and slowly they are using these programs in their creative process. While these programs are still crude, artists from the traditional media would just as soon consider them as computer graphics; yet I think that as they become more capable and flexible these programs will be absorbed into the domain of painting and drawing. In other words "computer graphics" as we know it today will disappear because of its success. Drawing on a computer will no longer be considered a separate medium but simply a perfectly common--possibly the most common--way to draw. Even in computer animation, the most spectacular form of computer graphics, most animations will soon be made with the aid of a computer, and therefore will be absorbed into the existing discipline of animation rather than remaining part of computer graphics. When we read a business letter, newspaper, or recent novel we know that it was probably written on a computer, but we don't think of it as "computer writing" because this practice is so common.

Photography presents another example. Today we generally think of a photograph as a print made through a silver based process, and we think of images digitized by a computer as "computer graphics." Artists who work with silver technology are "photographers" and those who work with computer and video digital images are "computer artists," even though both work with photographic processes. The distinction we are really making when we separate computer graphics from photography in this way is that photographs tend to be "seamless." That is, because of a photograph's high resolution and accurate color we tend to be most aware of the scene or subject represented, whereas a picture digitized by an inexpensive digitizing system (the kind artists normally use) has that "computer look" that reminds us of the process by which it was made. High resolution computer digitized pictures, on the other hand--such as those made by remote sensing spacecraft, are generally thought of as photographs. No one ever talks about the "computer graphics" Voyager sent back from Saturn. As high resolution digital imagery becomes less expensive, and even consumer snapshot cameras become digital, it will all

TH0164-4/86/0000/0027$01.00©1986 IEEE

be considered "just photography."

If the computer is accepted as a legitimate tool used within traditional media, will there still be a special category of "Computer Art"? I think there certainly will be, and that it will have a profound impact on the arts.

Alan Kay pointed out in his September 1984 *Scientific American* article, "Computer Software," that the computer is not a medium but rather a meta-medium. In other words, with the computer you can *create* media. A computer can be programmed to act like a typewriter, a calculator, a musical instrument, or drawing paper and pencil. Most applications for computers today use the chameleon-like quality of the computer to act like something that already exists. While the simulation is rarely perfect, the computer can add its own unique capabilities to the application. Rarely, however, do we see the computer used for really new purposes. It is here that the artist can make important contributions.

Artists have long been involved in exploration and innovation, and one of the primary concerns of contemporary artists has been to develop new connections between media. Some entirely new media have come out of such explorations: artists quickly understood the potential of video and have recently developed the new medium of performance art. In this context, what could be more exciting than the computer, the meta-medium, and who is better disposed to discover new applications than artists?

This is why artists should learn to program. If artists are going to create new media and develop new relationships between traditional media, they probably will do so through programming, because it offers artists the flexibility they need. Most existing software was modeled on some existing medium or application; what artists must do is expand what already exists. To do so they have to be able to create their own software.

What can computer programming bring to the Arts? The possibilities are limitless, but here is one simple example. Because of the hard work and enthusiasm of the last two department heads, Ken O'Connell, and David Foster, the Art Department at the University of Oregon has been active in establishing the School of Architecture and Allied Arts' computer graphics lab. Two of my students, Rodney Sitton and Alan Hartman, have been working with mathematics professor Richard Koch there to produce fractal images on our Vectrix graphics computer. Before their project it seemed to me that, despite their conceptual richness, all fractals looked the same. (Actually, since everyone was working with the same basic algorithm, except for cropping and scale they *were* the same.) This team's innovative approach to color changed my mind. Fractals are recursive, and a common way to color them is to assign a different color to each level of recursion. This group's fractals had over a thousand levels of recursion, and that's a lot of colors to deal with. They came up with the solution of using the computer to organize and manipulate colors, writing a separate program, "Color Wash," to do this. After the fractal was created, within seconds "Color Wash" could create new color harmonies based on mathematical relationships between RGB values. Many color schemes were tried and the best kept, yielding some very powerful images. It is important that in this situation the computer was used to manipulate colors in a way that would have been impossible for traditional media, given the large number of colors involved. The group used the computer to find a new way of dealing with color in the arts. Of course, this innovation would not have been possible had the artists not known how to program.

Artists and Software Development

Besides using the computer to create art, I think artists can contribute a great deal to software development. Despite the time and money committed by corporations and individual software developers, when it comes to developing new applications and new approaches to solving problems the industry shows a profound lack of imagination. Occasionally there are truly original products and innovations, but the vast majority of development effort seems to go into imitating other successful products, or to making some slight improvement on an existing application. With the tremendous power and flexibility of the computer available, why are there so few types of computer programs? Word processing, data base management, spreadsheets, games, and now paint and draw programs make up practically all of the program types people use. One could argue that there are a limited number of uses for computers, and that these have all been discovered, but this is clearly not the case. There are many applications for computers waiting to be discovered by anyone with the imagination to look. Before Dan Bricklin and Bob Frankson developed VisiCalc no one knew they needed a spreadsheet program; but now that the application is established millions of people have decided they can't get by without one. It is a sad comment on the industry that one of its major innovations in the past few years is integrated software, basically a repackaging of the standard computer applications.

I think that artists are ideal people to bring creativity and imagination to software development. Unfortunately, most people, including most artists, look on computer program writing as dull and unimaginative. When I tell people that I teach computers in an Art Department, many are as shocked as if I had said I taught auto mechanics in the English department. In fact, program writing and program design are creative activities; since programs must be both effective and imaginative, programming fits very well with the design concerns of many artists and art departments.

Teaching Artists to Program

I have been pleasantly surprised to find that most college students have had some experience with computer programming. Most artists, however, have not developed an active interest in it. A glance through an introductory level programming text book would convince most artists that programming is just as dull as they thought it was. Examples and problems in these books are taken from business, science, and math--rarely from graphics or art. There are good historical reasons for this, but computers belong no more exclusively to business, science, and math than they do to art. If artists are going to learn programming, then programming must be taught in such a way that artists can relate to it.

I propose that programming be taught as a studio course. Students should be quickly introduced to graphics commands, and all assignments should relate to the visual and conceptual possibilities offered by the computer. Artists are used to dealing with imagery in a tangible form and are often frustrated at being able to create only fugitive images on a CRT screen. Computer labs oriented to artists should be equipped with as wide a variety of hardcopy options as possible. Early in the course students should be taught how to make photographs from the CRT screen. Ideally, an arts-oriented graphics lab should contain high resolution color graphics systems, plotters, laser printers, film recorders, and animation facilities.

It is also important to introduce artists to the short but fascinating history of computer graphics. It has taken over a hundred years for the history of photography to be recognized and taught in art history courses, and it might take as long for the computer's contribution to be acknowledged. If we want this subject to be appreciated, we must make people aware of it ourselves.

The selection of the computer language in which artists learn to program can make the difference between a rewarding and an adversarial relationship with the computer. Fortunately, there have been some advances in computer language design in the past few years, and several new languages are much more friendly to first time programmers.

While many advanced graphics applications require the speed of a compiled language, very few, if any, first time programmers need this speed, and there are drawbacks to achieving it. For example, it is frustrating for the novice to go through editing, compiling, and linking, only to find that a simple syntax error was made. A long turnaround time is especially troublesome for the graphics programmer, because many times he or she wants to view the graphic several times as it is developed. Until a couple of years ago the only practical alternative to programs with these problems was Basic. For the most part Basic worked well, but because of its lack of structure and poor program formatting, longer Basic programs were hard to write and even harder to read. Recently we have seen several excellent alternatives. My current favorite is Macintosh Pascal.

Macintosh Pascal is an interpreter, so it allows users quickly to execute programs and view the graphic as it develops, while at the same time helping (or, rather, forcing) the user to structure and properly to format the program. It also catches simple syntax errors when the program is typed in. Macintosh Pascal offers easy access to the enormous number of graphics routines built into the Macintosh ROM toolbox. The biggest drawbacks to using this language are its slow execution speed and the Macintosh's lack of color, although at the University of Oregon we have used Macintosh Pascal to drive a Vectrix 384 color graphics processor. There is a similar product for the Apple II called Instant Pascal.

Basic is also getting better. Many new Basics, like the new Microsoft Basic for the IBM, Amiga and Macintosh, include some of Pascal's structure and formatting. They also use labels instead of line numbers and include a large number of graphics commands.

In the future we will see many significant improvements in computer languages. Object oriented languages like Smalltalk will be favorites for artists.

Conclusion

I want to observe, in conclusion, that the artist's responsibility is to discover and innovate. The artist is granted a freedom to explore ideas without being bound to the practical and the profitable that others in society are often denied. This freedom to create has an important role to play in the formation of computer graphics/computer art as a medium in its own right, and, incidentally, in the design of computer software in other fields. We are only just beginning to feel the computer's impact on our world. A knowledge of programming will empower artists to contribute to the imaginative use of the computer.

Controlling Animation with Data Files in the film, "The Journey"

Patricia Raun Craig

Department of Fine and Applied Arts
University of Oregon, Eugene, Oregon

In recent years, more artists have begun exploring computers as a tool for the creation of art. One art form that utilizes computers easily is animation. The film "The Journey" was produced using an IBM PC and Vectrix computers. In the production, the visual composition was only one problem that had to be addressed. Data files and data handling routines had to be written to deal with such complex problems as color changes, movement and shape.

Since the early 1960's, artists have been collaborating with engineers and scientists to create new forms of art. These collaborations were many faceted and included environmental happenings, light sculpture, large-scale sculptures, and video and computer experiments. While computer graphics programming systems were being developed, artists were involved in experimenting and pushing these systems to their limits.

Cranston Csuri received a grant from the National Science Foundation to develop user interface devices such as the light pen. John Whitney became the "artist in residence" with IBM computer systems, where he gained access to equipment and information of the most advanced kind. Lillian Schwartz worked with Ken Knowlton at Bell Laboratories to develop the first computer "paint" systems. Ivan Sutherland, one of the early developers of a "sketchpad" system, was advancing hardware capabilities and taking the first steps into the realm of three dimensional space.

As the software and hardware developments grew more complex, computer scientists became more involved with computer graphics. With the additional influx of new scientists, the technical aspects of software and hardware advanced to include the capabilities of texture mapping, surface reflection, and the imaging of fuzzy surfaces such as clouds. While computer graphics became more technically oriented, the number of artists involved with this new medium did not greatly increase. Many of the films and images developed on computers became "cold" and mainly concerned with the wonderful technical capablities of a specific software system. The images were generally void of meaning and concerned only for a realistic appearance of the objects depicted.

Slowly, in the 1980's, more artists started to explore computers as a tool for creating art. Artists gradually became fluent in computer software tools such as paint programs and graphics programming. No longer will the computer-generated art be a mere "cold" esthetic picture; rather, it will be an art infused with inventive and expressive meaning. An explosion of new imagery will emerge and artists will use the computer for expressing their world views. The purpose in creating an animated film is to move the artistic creation on computers in this direction.

A variety of artists (e.g. animators, printmakers, sculptors and photographers) can use the computer in the art- making process. The computer can create the in-betweens for the animator, while for the sculptor it can help calculate stress and intersection points. Digital photography is made possible through interfacing a video camera with a computer. Printmakers and painters can use the computer as an image generating device. Various output devices such as ink jet printers and plotters can be used as new printing presses. Any artist who uses a planned, analytic approach is well-suited for incorporating the computer in the art making process.

Through printmaking an analytic approach to the creative process is developed. With each print there is delayed gratification, and often the artist is uncertain of the progress towards the final product. A painter knows the result of his actions immediately, but the printmaker makes many decisions before a proof is taken. Filmmaking has a close kinship to this method of working because an animator plans and develops working drawings for long periods before his work is viewed.

My primary purpose in the animated film, "The Journey", is to express the personal story of my Christian rebirth. Since the experience was a slow awakening

TH0164-4/86/0000/0031$01.00©1986 IEEE

rather than a dynamic emotional event, the film is a journey which continues slowly and steadily. While the film progressed, a hierarchy of concerns developed which clarified the film's vision. In the development of the story board, symbols were included or removed based on communicative value and underlying meaning of each symbol. Choices dealing with composition, design detailing and movement were made based on contextual meaning of the film. The inclusion of sophisticated programming techniques was based solely on compositional problems. By prioritizing the elements of the filmaking process, it was possible clearly to base all programming decisions on the ultimate goal of communicating my rebirth experience.

As a printmaker, the compositional process of using the collage was pervasive and this thought process influenced the production of the film, "The Journey". Overlays of various types of movement were collaged together to develop a diverse range of motion. The film dissolves are successful because they are moving and active transparencies of the dance. An artist's personal symbols can change meaning based on surrounding references within the collage method of working. Rauschenberg's work is a great example of transforming the meaning of symbols within the scope of his work. The lotus flower moves from an element which symbolizes death into a vibrant object of freedom through the final appearance of the flower as an illuminated bird. The ability to preceive various layers of meaning was made possible by incorporating the collage methodology.

Since the Israelites' Exodus from Egypt can be viewed as an allegorical story of a Christian's departure from slavery and death into a life of freedom, culturally significant patterns of the Egyptian lotus and the Tibetian symbol of the eternal knot symbolize my journey. At the beginning, the Egyptian lotus are seen as a static wallpaper pattern, lifeless and in dark values. Slowly the lotus stem grows in darkness and, through touching, awakens a nearby lotus flower. Mysteriously, the large leaf and stems transform into beautifully pastel shapes and slowly glide across the screen, finally disappearing. Concurrently, the awakened lotus flower begins breathing with movement similiar to human lungs. The static two-dimensional wallpaper pattern dissolves into a free flowing dance movement on a flat plane. Identity with the warm-blooded world grows as the background changes from darkness into a warm fabric coloring. The environmental lighting seems to become stronger as the stems hover over the screen as shadows and deepen in value.

With the arrival of the eternal knot, an object of well- being and good luck, there is more life and energy expressed in the activity. The eternal knot twirls and tumbles freely from a central location bringing images of butterflies to remembrance. Since the butterfly's life begins as a crawling caterpillar, it is a symbol of new life and resurrection. Because of the singularity of this object, it creates a feeling of empathy and reference to the self.

As the fluttering butterfly increases our awareness of a deeper space, the dancing diversifies and broadens references to emotional and intellectual life. The lotus flower and eternal knot sway away from rapidly flying stems. The lotus flower bends down or turns toward its side as it rotates in a three dimensional plane. At times, the eternal knot symbol hestiates and slows as it travels through a path of rapidly flying stems.

References to resurrection imagery are multiplied through a crown of thorns which becomes a joyful reminder of Christ's death. The sharp and stylized stems are rotated in an elliptical orbit, through the movement visually conveying the symbol.

Skeletal forms from the lotus flower become bird imagery symbolizing freedom with references to purity and peace. The birds are transparent and are seen as illuminating the surroundings with light. Their flight is seen as a joyful minuet as they join together forming new shapes and colors.

Symbols of purity are enhanced through color, awakening images of water and cleansing. By converting the background color from a warm nuturing color through a process of slight darkening and finally infusing light tints of blue, an environment reminiscent of water is obtained. The thorn imagery floats through space and recalls visions of swimming schools of fish. Finally, while the objects continue to dance, the colors fade into a clear sky blue and the ending titles appear. The continuation of the dance is symbolic of the never ending journey of our lives.

The development of software routines for handling movement, acceleration, lighting, and color effects were integral for the development of the film. These routines were generalized procedures which handle plotting and movement data. The flexibility of using data files was primary in the decision-making process. The vast amount of data required an immense memeory or the development of a data base file system on disk. Since each lotus initiates its movement at different intervals, during any individual frame the specific fifteen lotus flowers will be transforming at varying rates and various stages. Since all the movements are completed by the use of data files, the management of these files became a necessary task during this project. Retaining proper backup copies of these files became necessary. The generalized routines will be described specifically in reference to the technical aspects of the film.

For the initial sections of the film, a general file controls the tempo for the initiation of each movement, which is called shape.dta file (Pascal version is called nshape.dta). The variable f_btfly determines the activity type of each lotus shape, such as the growing or breathing movement. In addition to this file variable, there are two program arrays which hold the counter and the limit for each value of f_btfly. Obviously, when the counter reaches the limit for the current value of f_btfly, the file is changed to reflect these changes, thus adavncing the movement.

Since the first moving objects must appear as though they are in front of the other objects, the first moving objects must be drawn last. Objects which are drawn last on the computer screen are always forward of all other objects. Nxt_shape variable in the shape.dta file prioritizes the drawing sequence for the lotus shapes. Hence, the screen is not drawn right to left as most wallpaper patterns but from the last moving lotus shape to the first. By starting shapes in a non-arbitary manner, the viewer is surprised by new activity on the screen. By using one variable to control the drawing order, the computer will have a consistent method for the drawing order of all objects. One object will not suddenly appear forward of another object, and visual continuity will be retained. The data structure for shape.dta is found in Table 1.

Flw_ite points to the record number in the flower.dta file (Pascal version is called nflower.dta) which holds specific data concerning the lotus's breathing and dancing routines. Flw_x, flw_y, flw_z are the actual screen placements for each lotus flower. Two pointers are needed for the stem.dta file (Pascal version is called nstem.dta). Stm_ite is the first related record in the stem.dta file and is used for the center stem and large leaf. Rl_shp references the stem.dta through accessing a related record in the shape.dta and is used for the left and right stems. Since another object's stems cause the current object to awaken by a tickling movement, the current object needs to know which object's stems will stretch, touch and tickle it. For example, for object #5 to start moving, the stems for object #1 need to grow and touch object #5.

After the screen colors fade to darkness, the movement commences with the left or right stem growing so they touch a nearby lotus flower. This action causes the lotus flower to awaken and change the colors of its petals. The growing movement is accomplished by changing the values of the y-coordinate and gradually changing the x-coordinate for stretching. One hundred polygons compose each stem while it is growing for a smooth silhouette. For a speedier drawing time, there was a faster plotting routine for the non-growing stems.

After the stem grows to the appropriate height, it will appear to tickle the lotus by swaying back and forth. This movement should be reminiscent of grass moving in a gentle breeze. This effect is accomplished by changing the value for the x-coordinate.

After the growing movement is complete, the stem transforms into a new variety of stylized stem with only three basic polygons. The method for the metamorphosis is similiar to the growing and breathing movements. The metamorphosis is handled through variables which cause expansion of the sides and creates the texture. As soon as the metamorphosis is complete, the stylized stems are ready to begin their rhythmic movement as individual elements. As the film proceeds, the next type of movement is the lotus flower breathing patterns. The lotus flower gradually breathes deeper and faster as it awakens. The flower.dta (Pascal version is called nflower.dta) and the grow.dta (Pascal version is called pgrow.dta) are the files which handle the breathing data. The grow.dta holds the exact placement of

VARIABLE NAME	CONTAINS
NXT_SHAPE	next related shape record number
FLW_ITE	pointer to flower.dta file
STM_ITE	pointer to stem.dta file
F_BTFLY	flag for starting next lotus
RL_SHP	another pointer to stem.dta file
FLW_X	translate for X-axis
FLW_Y	translate for Y-axis
FLW_Z	translate for Z-axis

TABLE 1: Data structure for shape.dta file.

VARIABLE NAME	CONTAINS
GRW_ITE	pointer to grow.dta file
SSHAPE	pointer to shape.dta file
CLR_ITE	pointer to color.dta file
MV_REC	pointer to stem.dta file
NEW_LOTUS	pointer to stem.dta file
DISAPPEAR	amount of frames lotus disappears
FLW_RX	rotation for X-axis
FLW_RY	rotation for Y-axis
FLW_RZ	rotation for Z-axis

TABLE 2: Data structure for flower.dta file.

polygon points. For each flower.dta record, there are seventeen records in the grow.dta, since seventeen polygons compose the lotus flower. The structure for the flower.dta file is found in Table 2.

Within each file structure, the variable sshape or rl_shape will trace the record to the shape.dta file. In order to manage the color changes, a color file is accessed through clr_ite. The current colors are stored in the grow.dta file, but the final color values are stored in the color.dta file. All the current values for rotation are held in flw_rx, flw_ry, flw_rz.

Grw_ite points to the grow.dta file which will access the remaining sixteen records for the lotus flower. Table 3 contains the data structure for the grow.dta.

Since all the relative coordinates are accessed from the data files before drawing the lotus flower on the screen, the fsect becomes the value for the array of seventeen polygons. The fsect was also useful during the debugging process. Nxt_flw references the next related record in the grow.dta file. Since there are four colors in the lotus, it was necessary to retain this information in the data file. The colors in the lotus flower change at varying times during the flim, so each individual object needed a different set of color numbers. The series of coordinates for x and y axis are for the placement of the lotus during the breathing process. Mv_x and mv_y are relative move commands for the precise drawing of the lotus and tex_x and tex_y are used for the relative moves while drawing the texture. The actual breathing is handled by increasing or decreasing the x and/or y coordinates. With relative coordinates, when one coordinate is changes, another corresponding coordinate must change in the opposite direction. Only the acute and obtuse positions were plotted and the computer calculated the intermediary positions. By gradually changing the final acute or obtuse movements, the lotus flower appears to awaken and with increasingly deeper breaths anticipates its dance.

Gradually all the shapes initiate their free flowing dance movement, gliding across the screen. Each dance was individually plotted and finally combined to form a visual representation of the overall movement which was planned and correctly anticipated. All the

VARAIBLE NAME	CONTAINS
FSECT	flower or polygon section number
NXT_FLW	pointer to grow.dta file
SSHAPE	pointer to shape.dta file
F_COL	color number for current polygon
F_RD	red value for current polygon
F_GRN	green value for current polygon
F_BL	blue value for current polygon
X1	first relatine x-coordinate
X2	second relative x-coordinate
X3	third relative x-coordinate
X4	fourth relative x-coordinate
Y1	first relative y-coordinate
Y2	second relative y-coordinate
Y3	third relative y-coordinate
Y4	fourth relative y-coordinate
MV_X	relative move x-coordinate
MV_Y	relative move y-coordinate
TEX_X	relative move x-coordinate
TEX_Y	relative move y-coordinate

TABLE 3: Data structure for grow.dta file.

movement was visually represented at one second intervals as per the diagrams in the back of the report. While debugging the program a variable called "add" was used to simulate one second, half second or two second intervals. This variable was added or multiplied by each movement variable, which effected all changes and transitions during the film.

Breaking away from the static wallpaper pattern, the lotus flower movement is handled through show, mv_rec, new_lotus, and disappear variables. Mv_rec and new_lotus are the variables for the first related dance-like movements in the stem.dta file. The dance-like movement begins with a swinging from one screen edge to its opposite and clockwise to counterclockwise rotational movement. The new_lotus and disappear are used when the lotus flower is moved closer to the viewer and then gradually diminishes in size. When the new_lotus variable is accessed, rotations on the x and y axis are changed to depict the lotus slanting into a three dimensional space. As the lotus flower looms closer to the foreground, it will dissappear for a specific number of frames which is controlled through the disappear variable.

All the break away movements (for the large leaf, left and right stems, the lotus and the eternal knot) are handled through the same procedure which accesses the stem.dta file. This is the largest file in the data system and contains 400 records. The data structure is found in Table 4.

Just as sshape and rl_shape traces the object to the shape.dta file, rl_rec traces the object to either the stem.dta or flower.dta file and adds to the clarity of the data structure. Nxt_move refers to the next related record in the stem.dta and contains additional data for the dancing movements. Mv_type is the variable which determines the kind of object which will be drawn.

The variables (enx(1..3) and eny(1..3)) were used to determine the final translate values for the x and y coordinates. When the current value for the x coordinate is equal to or greater than the ending value, then the next ending value is used for determining the direction of movement. To create a greater variety in the type of movement, it was necessary to program a method for handling access to an ending x value while retaining the ending value for y. Dir_val has a initial value of 11 and may become as great as 33. The following diagram explains how this variable is used:

DIR_VAL	VARIABLE
11	enx(1),eny(1)
12	enx(1),eny(2)
31	enx(3),eny(1)

Stm_mx, stm_my, stm_mz hold the values for translating the lotus on the x,y, and z axis and represent movements for each frame, whereas s_rot holds the rotation on the z axis.

With the arrival of the eternal knot symbol, the method of handling translation is changed from actual values to a system which determines the movement based on screen percentages. Since the action occurs in a three-dimensional world and the objects advance and recede greatly in this world, actual coordinates would not accurately reflect the dancing movement. Relative values based on the percentage of the screen are employed for initial, ending, and current coordinates for the lotus and eternal knot objects. When the mv_rec variable is empty, the new_lotus variable is accessed, thereby utilizing the screen percentage procedures. Based on the z-translate value, the screen percentage values are analyzed and actual coordinates are calculated. These routines allow a wider diversity of action, enabling the environment to move from the flat plane to a three-dimensional world.

VARIABLE NAME	CONTAINS
RL_REC	pointer to related files
NXT_MOVE	next related in the stem.dta file
RL_SSHAPE	reference for the shape.dta file
MV_TYPE	type of object (e.g. stem, lotus)
ENX(1)	first ending x-translate value
ENX(2)	second ending x-translate value
ENX(3)	third ending x-translate value
ENY(1)	first ending y-translate value
ENY(2)	second ending y-tranalate value
ENY(3)	third ending y-translate value
STM_MX	amount of x-movement
STM_MY	amount of y-movement
STM_MZ	amount of z-movement
S_ROT	amount of z-rotation
DIR_VAL	array value for enx,eny

TABLE 4: Data structure for stem.dta file.

In order to maintain the illusion of a spatial environment, when the eternal knot object moves forward and the lotus flowers recede, an insertion sort procedure determines the order of drawing for each screen. After each object's translations and rotations are calculated, these values are inserted in an array based on the z-translate value. The lowest z value is drawn last in the drawing sequence, since it is the largest object, thereby abiding by the spatial perception laws. For drawing purposes, in addition to the translate and rotation values, color values (including the red, green, and blue values) and the mv_type are stored in the drawing array. The mv_type will access the correct plotting routine for the lotus or the eternal knot.

Since the activity of the objects advances and recedes, collisons are probable. Collision procedures checked the current values of the lotus against an array of stored values for the eternal knot symbols. When the lotus translate values are within a certain range and the objects crash, a flag is set so the eternal knot's colors will shimmer and glow. The probability of a crash was high, however all the collision procedures were programmed in vain.

As the film progresses, the dance movement is plotted through an analytic geometry function, the ellipse. Various sizes and dimensions of elliptical orbits were calculated, and the initial coordinates were placed off-screen. While the stylized stems rotate around this orbit, a visual reference to a crown of thorns should be possible. To make this section more successful the pathways should be plotted more precisely so the ellipse would be more visible. Since the elliptical orbit is larger than screen boundaries, instead of rotating around an invisible oval shape, the stems seem naturally to glide across the screen, appearing from outside the viewing area. The individual stem's passage depicts the thorn imagery; however, due to the abundance of stems, it is impossible to follow each individual stem. This section becomes an overall pattern of great activity.

COLOR HANDLING

Throughout the film, color is used to describe transitions. There are 512 addressable colors out of a possible range of 16.8 million colors on the Vectrix computer system. Since there is a command for changing a specific addressable color to a new chroma, hue or saturation, the ability to use color to describe transitions is greatly enhanced. Instead of using lab techniques for dissolves, the computer is used to dynamically control dissolves. The initial light background darkens to black, then lightens to a warm beige and then darkens to a medium gray and finally lightens to a light blue. This is accomplished while the objects are moving and changing. All the objects at specific points will transform from dark, grey colors to warm, brighter colors, thus signifying a transition from death to life.

The Vectrix computer system has various color function routines which can be used to create shadows and illumination effects. In order to create the effect of expanding brightness, methods for handling shadows and illuminations were created. Normally, in order to display color on the Vectrix computer system, the opaque color mode (re) is utilized. When using the replace complement color mode (rc), the Vectrix ignores any previous color command. After determining the current screen color, the Vectrix computer will subtract this value from 511 (represents the total number of colors) and draw the object with the new color value. By dividing the addressable colors in half and using only 255 colors to draw individual objects, the remaining 255 colors will store darker versions of the initial 255 colors. Hence, if color 10 is used to draw a stem, the shadow color will be 501. By putting red, green and blue values of 100,100,150 into color location 10 and values of 70,70,120 into color location 501, a shadow may be drawn using the replace complement color mode.

The illuminations were created using a similiar method through a simulated transparent color mode (or). This color mode will add the current drawing color to the screen color and create a new color value. This new color value is used to draw the object on the screen. Hence, if the screen color is 100 and the current drawing color is 50, the actual drawing color will be 150 when using the simulated transparent color mode. By placing a lighter version of color 100 into color location 150, an illumiantion effect is obtained. By putting red, green, and blue values of 100,100,150 into color location 100 and by putting values of 150,150,200 into color location 150, an illumination may be drawn with this color mode.

The lettering is in the font style of the Bodoni typeface. In order to use this font style, individual letters were plotted with numerous polygons. For example, the letter S has about 100 polygons. By plotting the letters in three-dimensional coordinates, a typeface with great utility was employed. In the opening sequence, letters fly into place in an orderly manner. The movement does not simulate a corporate logo fly-by but prepares the audience for the noble dance. In addition to creating the letters, an algorithm for the proportional placement of letters was created. This algorithm eased the development of the ending title screens. Since the program is written in Basic, it enables an interactive appraoch for determining correct placement of titles. Another short program was written to fade the titles from sky blue to black and return to the sky blue through red tones. This method of fades leaves an unusual after image, thus assuring the complete environmental control of the film.

Throughout the programming process, the ultimate goal of communicating the concept of the rebirth experience was constantly referenced. All of the programming decisions were not based on the frills of the Vectrix computer system or displaying a certain expertise in programming but rather were always based on the content of the film. The filmmaking process is an excellent means to capture an audience and hold their attention. The problem of capturing a viewer's attention has always been one of my primary concerns, and I believe that film may be the correct medium for my future work.

Desktop Publishing: An Introduction

Frank Kulesa

Northern Illinois University

Abstract

This paper introduces electronic desktop publishing, which is currently being taught on the undergraduate level in the School of Art at Northern Illinois University. It is mainly written for educators who are interested in, but do not have any knowledge of desktop publishing. I hope it will serve as a beginning, showing why desktop publishing is desirable within a graphic design environment.

Introduction

Electronic page layout borrows many principles from computer aided design and traditional drafting, enabling a graphic designer or layout artist to combine text, art and halftone photographs into a finished document.

Due to constant improvements in microcomputer hardware, software and disk memory space, new page layout programs are arriving onto the microcomputer world.

Early mini or mainframe systems had a price tag of over \$100,000. Now some stand alone mini and micro systems are between \$40,000 to \$80,000. These systems are directed to professional designers that want very high resolution and color in the finished output.

Due to processing and memory advances in the microcomputer, combined with their wide use in business, page layout programs are becoming popular. Although these programs are not as sophisticated as larger systems, several do perform a wide range of page design functions for printing tabloids, newsletters, small magazines, company reports, or any type of document a company feels is appropriate.

Most page layout manufacturers have decided to support the IBM PC and the Apple's Macintosh. Although the IBM PC is more widely used, its graphic capability has to be improved by a third party. Unfortunately these third party systems have no "standard" product between them. While the Macintosh may not have the calculating power of later PCs, it has one of the more powerful and best interactive graphic capabilities for a microcomputer. Software supporters have taken advantage of Macintosh's friendly interface, developing a family of products with few compatibility problems.

Both IBM and Macintosh page layout programs let the text and graphic creation be achieved on separate word processing, "paint" and "draw" programs. Art can also be generated from digitizers and scanners. By opening and closing these files while in the page layout program, the text and graphics are positioned together to form a finished document. Some programs will output what appears on the screen, while others show a mockup with block graphics and text. Probably the most important feature is the ability to interactively design and edit the document on the screen, which radically reduces production time. Changes on one page will automatically update parts of the rest of the document. After the document is completed, the information can be sent to a number of printing devices. These devices range from dot matrix printers for "proofs" to a laser printer or professional typesetter for a camera-ready hardcopy, or as the final output. For those that do not want to invest in output devices, some programs allow the document to be sent by modem or disk to a printer or typesetter.

The best software mimics the traditional drafting table of the layout artist. Text and graphics can be placed directly on the page or stored off to the side for later use. All layout tasks are handled directly on the screen, this improves the ease and quality of the layout. Most microcomputers have small screens, therefore the ability of zooming the document to different sizes is needed.

It is important to remember that the basic rules of page design and aesthetics have not changed, just that new tools have been added. These new tools will help most people solve their publishing needs. What type of publication, who the audience is and for what purpose determines, the output.

Why Desktop Publishing

In order to keep our students current with the expanding area of printing documents inhouse, a course was developed to introduce the techniques and advantages of desktop publishing. As the inhouse publishing field grows, so will the need for more trained people who can operate the computerized systems. These new designers will have need of typing and word processing skills, an appreciation of good digital typography, and mastery of graphic and publication software.

In simple terms, desktop publication uses the microcomputer to write, design and print documents.

There is a mass increase in the amount of availability of information, how it is stored and handled. Text and graphics are now transformed from paper into the electronic medium, which is changing the traditional materials of the graphic designer. This causes for some an uneasiness, because older and familiar design methods and materials are transformed into electrical signals. Designers that are comfortable with traditional page layout methods sometimes find it hard to learn new skills and concepts, and to accept the computer as a viable design tool.

Inhouse publishing is not new, anyone with a duplicating machine or printer can publish. What is of interest, is how to produce documents that look professional

TH0164-4/86/0000/0039\$01.00©1986 IEEE

at a low cost.

Now that printers and typesetters are computerized, text and graphic alternations are easily accomplished. Combining innovative design capabilities with precise on screen editing of text and graphics, small publishing jobs can look professionally composed and printed. Gone are long periods of writing, editing, typesetting, cutting and pasting of mock-ups, and coordinating with the printer. Instead of paper, rubber cement, press type, composers, drafting table, etc., the designer now works with digitized text and art, different screen formats, hardware, software, interfaces and how computerized information is transferred to the printer or typesetter.

The main advantage of electronic desktop publishing is the sophisticated production aids it gives a designer in producing publications at a speed that traditional methods do not allow.

Desktop Publishing and the Designer

Producing effective visual communication is the job of the graphic designer. Although desktop publishing allows the designer to design documents faster, it is his responsibility to produce a document that communicates effectively, and to communicate, a publication must have clarity, order, and meaning.

The basic design process does not change when working with the computer and publication software, (Definition of the Problem, Analysis, Alternatives, Solution, Production and Evaluation). The computer can be useful in the total design process. With publication software, the process of finding a design solution is impressively shortened. During the solution process, greater flexibility for generating ideas and their alternatives is permissible, changes can be done up to the last-minute before printing. Repetitive hand labor activities are computerized, making the design process more economical, allowing for a wider spectrum of ideas. A layout can be rapidly modified and reviewed, which means more ideas can be tried in a shorter amount of time than using traditional methodology, hopefully to produce better solutions.

In the beginning, there is the risk of allowing the technology to influence the output, but a good designer will use the computer as only a tool, not as "a means to an end."

Basic Software Needed

Desktop publishing software have very limited text processing capabilities, therefore a word processor is needed to write and set text. For producing art, "draw" and "paint" programs are required for ample graphic tools. If business charts and graphs are generated in large numbers with constant revisions to data, then a business graphic package would be valuable. To capture three or two dimensional images, a digitizer or scanner with graphic tools would be needed. Most inexpensive digitizers and scanners do not produce the halftone quality of photographs. If photographs are used, locations are reserved in the document, later the photographs can be placed into these locations before sending the camera ready document to a professional print shop.

If large or numerous small documents are produced, disk memory is quickly filled. To eliminate the nuisance of removing and replacing file disks, a hard disk is needed. Also the new double sided internal and external disk drives are helpful to eliminate "disk shuffling".

Spelling Checker Software

If your word processor does not correct spelling, a spelling checker is an important addition. Checkers can work as accessories or as a stand-alone addition. They can review a whole document or a block of text, others are interactive, looking for errors as you type. The size of its dictionary will influence its speed and memory capabilities. Be aware that the size of the dictionary can be misleading. Some manufacturers count root or variations of root words, omit common words, flag contractions or plurals as errors. Others will allow new words to be added to its dictionary. Special features are available; thesaurus, glossary, proper hyphenation, but the most important feature is how fast it catches misspelled words and how few correct words it flags as a misspelling.

Font Software

If the supplied software and printer fonts are not enough to handle design needs, downloadable fonts (temporarily loaded) can be added to the system. Building a large font library can be a serious investment, but by mixing font sizes and styles of just a few fonts, many text variations can be created.

In choosing a font, one should look at its aesthetic quality, ease of use and especially its design versatility.

Some Basic Font Faces

Text faces (*serif*)

With these faces the eye is able to scan the document quickly, also the typefaces have tapering strokes that are terminated with serifs that allow for easy recognition of all letterforms. Text faces are used for lengthy passages of running text.

Display faces (*sans serif*)

These typefaces have almost uniform letter thickness which can create readability problems over a large area of text. *Sans serif* faces with light strokes can cause printing errors in laser printers when using some point sizes. Some pixels are removed during printing, changing the appearance of the text. Display faces are used in short passages such as titles, letter heads, cover design. The eye is attracted by their bold and dramatic style. Readability is taken care of by the type's page placement and its larger face size.

Decorative faces

These are the least used, but they serve the special function when the type itself is needed as part of the narrative. Decorative faces are too distinctive to blend well with informational typefaces.

It is important to choose fonts that can perform different functions. A display font can sometimes serve as a text font and a text as a display font. Choose a font style that comes with a large number of point sizes. Remember that a font's stroke weight should be consistent, the alignment of characters should be precise and all letter spacing should be even. Examine the size ranges, what looks good in one size, may not in another.

Printing time between typefaces vary, especially if you have downloadable styles. Also some downloadable fonts take up a lot of memory which slows down printing time or can cause crashing if memory areas are overlapped. Future printers with larger memory areas will solve this problem.

Copy protection can sometimes be a problem if the fonts are only imprinted to a particular printer. Also it may cost more if an on-sight license is needed for multi-users of a system.

Publishing Software Buying Checklist

Certain basic items should be looked for when purchasing a good publishing program. The designer working with the software should be comfortable with its operating procedures. Try the software out before purchasing, do not rely on someone elses opinion.

Text Processing
The software should allow for fast opening of text documents while in the page layout program. What options (deleting, adding, changing text, setting tabs, moving one word or whole blocks of text) are there to format text? How does the program accommodate multipage edit changes, does a change on one page effect changes on another? Can text be wrapped around a variety of shaped graphics? New publishing software will most likely contain better word processing capabilities.

Graphic Processing
The software should allow for fast opening of graphic documents while in the page layout program. How are graphics handled for cropping and resizing images? What type of drawing tools (different size paint brushes, erasing, zooming, etc.) and line variations (size and texture) are available? Do all the graphic tools work in all display sizes? Is there the ability for adding special graphic effects?

Page Processing
Can master pages and templates be created? How easy is it to insert, delete and swap pages? It is helpful to create and print several thumbnails on one printout. Does changes on one page automatically cause changes to the rest of the document? Must you save the whole document, or can you select certain pages to be saved?

Performance
What is the printing speed when text and graphics are mixed? How fast can documents be opened? How does your working style agree with the program's interactive operating design? Is the working style similar to traditional page layout methods? Is output page size limited or is there the ability to choose a variety of printing sizes?

Display
The screen in actual or expanded zoom display should show exactly what will be printed. Are there different zooming capabilities for accurate placement of elements? Can you do all the needed editing, placing, sizing and cropping on the screen? Are there guides and snap-to lines for controlling element placement? Can ruler measurements be set to points and picas allowing for photographs to be dropped into the printed page? Some displays will show ruler measurements in inches, centimeters, points and picas.

Hyphenation and Justification
Some programs have manual and automatic hyphenation. Justify, aline center, aline right and aline left is important.

Leading
Some programs have the ability to adjust line spacing in points on any line, allowing for more accurate spacing within columns.

Kerning
The ability to control spacing between letters and words allows for professional looking documents.

Hyphenation, justification, leading and kerning will be available on most quality page layout software in the future.

Support
What are the warranty terms? How much time is allowed to learn the software? What are the copy right laws and are the disks copy protected? Is the supporting documentation easy or hard to understand?

Transferring Information
The system should operate with an adaptable communication language. This allows documents to be transferred to different printers and typesetters from disk or across modem.

Length of Document
Some software only allow one page to be produced, others will generate enough pages to publish small magazines.

Copyright Protection
Different software products have a variety of restrictions. Some allow their disks to be copied, others do not. Usually the more expensive software will need a master disk to start the program. On-sight license may be needed for multi-user systems.

Examples

What follows are several student examples. The software programs used were: Apple's MacPaint (paint program), MacDraw (drawing program), and MacWrite (Wordprocessor). Also used were Thunderware Inc.'s ThunderScan (scanner software for capturing two-dimensional images), and Aldus' PageMaker (publication software).

Figure 1 is a diagram of the hardware system used to create and print the student examples. The computer is a 512K Macintosh computer with two single sided disk drives. The dot matrix printer is Apple's ImageWriter and the Laser printer is Apple's LaserWriter (300dpi). If numerous documents are printed, a Macintosh computer with a 10meg internal hard disk drive is used.

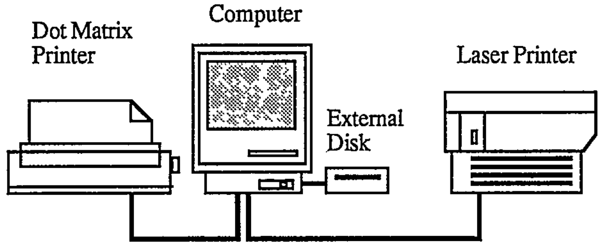
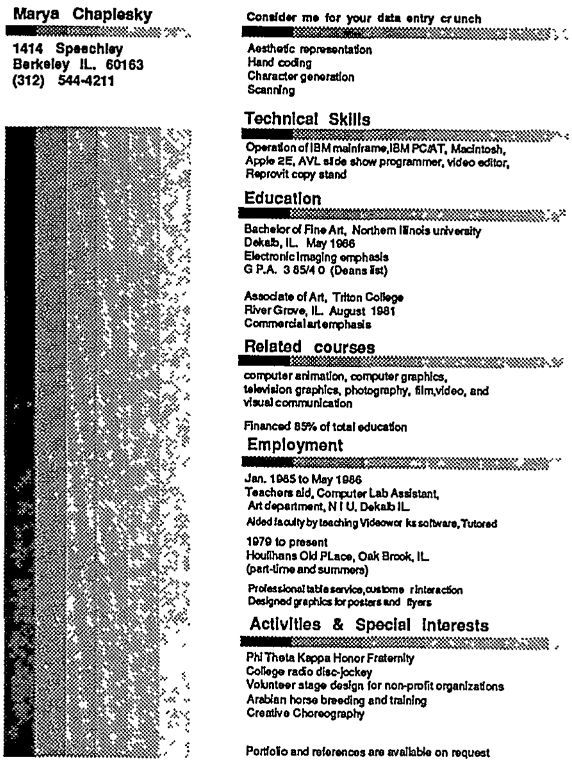


Figure 1.

Figure 2 and figure 3 are projects that did not use publication software. Both Figures were created with MacDraw. This program allows text and graphics to created and positioned on the same page. Documents can be created without publication software, and although the documents take longer, and are not as easy to create, layout work can still be accomplished.


Marya Chapiesky

1414 Speechley
Berkeley IL. 60163
(312) 544-4211

Consider me for your data entry crunch

Aesthetic representation
Hand coding
Character generation
Scanning

Technical Skills

Operation of IBM mainframe, IBM PC/AT, Macintosh, Apple 2E, AVL slide show programmer, video editor, Reprovit copy stand

Education

Bachelor of Fine Art, Northern Illinois university
Dekalb, IL. May 1986
Electronic imaging emphasis
G.P.A. 3.85/4.0 (Deans list)

Associate of Art, Triton College
River Grove, IL. August 1981
Commercial art emphasis

Related courses

computer animation, computer graphics, television graphics, photography, film, video, and visual communication

Financed 85% of total education

Employment

Jan. 1985 to May 1986
Teachers aid, Computer Lab Assistant, Art department, N I U. Dekalb IL
Aided faculty by teaching Videowor ks software, Tutored

1979 to present
Houlihans Old PLace, Oak Brook, IL
(part-time and summers)
Professional table service, custome r interaction
Designed graphics for posters and flyers

Activities & Special Interests

Phi Theta Kappa Honor Fraternity
College radio disc-jockey
Volunteer stage design for non-profit organizations
Arabian horse breeding and training
Creative Choreography

Portfolio and references are available on request

Figure 2. Resume, Marya Chapiesky, Northern Illinois University

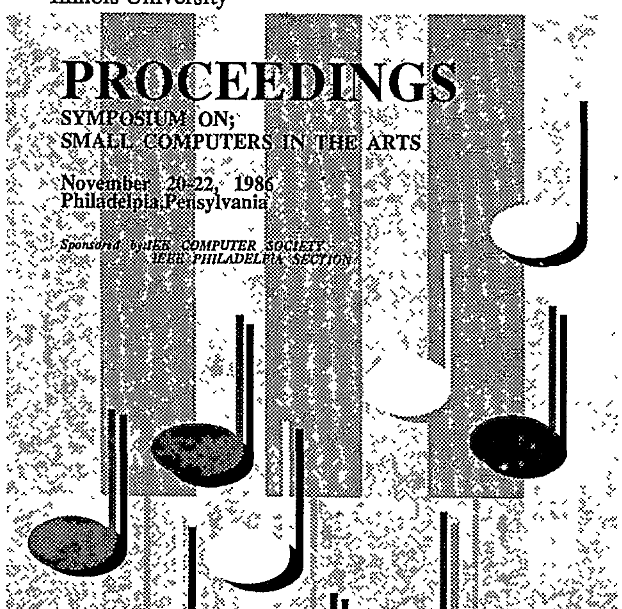


Figure 3. Cover, Michael Zeman, Northern Illinois University

Figures 4 through 7 were completed with a variety of programs for producing the text and graphics: MacPaint, MacWrite, MacDraw and ThunderSan. The text and graphics were then layed out and printed with PagerMaker software.

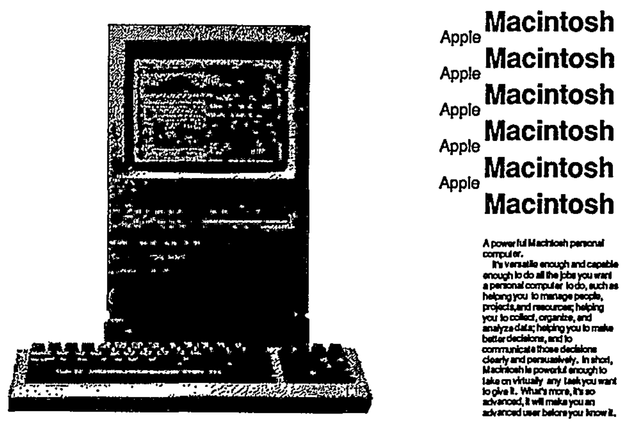


Figure 4. Brochure (page 1), Marya Chapiesky, Northern Illinois University

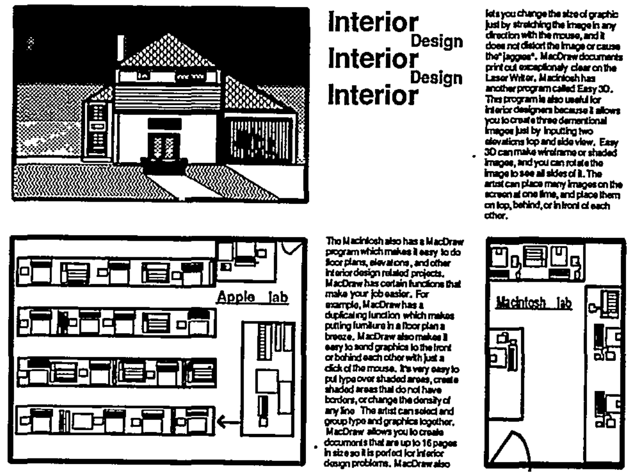


Figure 5. Brochure (page 2), Marya Chapiesky, Northern Illinois University

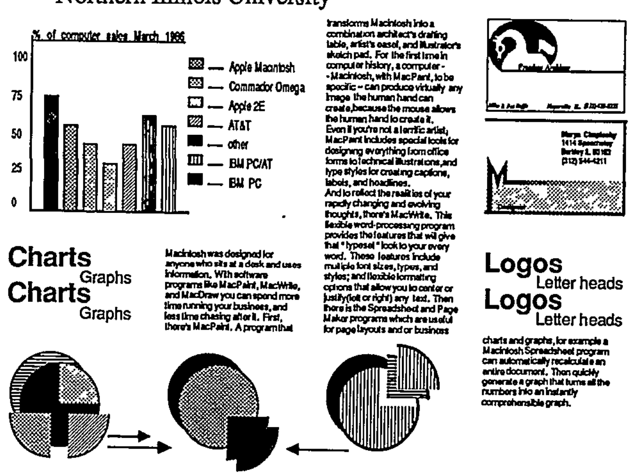


Figure 6. Brochure (page 3), Marya Chapiesky, Northern Illinois University

Figure 7. Brochure (page 4), Marya Chapiesky, Northern Illinois University

Page Layout Manufacturers for the Macintosh

PageMaker	**ReadySetGo**	**MacPublisher**
Aldus Corp.	**Manhattan Graphics**	**Boston Software Pub.**
616 1st Ave., Ste., 400	163 Varick St.	19 Ledge Hill Rd.
Seattle, WA 98104	New York, NY 10013	Boston, MA 02132
(206) 467-8165	(212) 989-6442	(617) 267-4747
$495	$125	$99

Page Layout Manufacturers for the IBM PC

SuperPage	**Do-It**	**Clickart Personal Pub.**
Bestinfo	**Studio Software**	**T/Maker Graphics**
33 Chester Pike	17862 Fitch	2115 Landings Drive
Ridley Park, PA. 19078	Irvine, CA 92714	Mountain View CA. 94043
(215) 521-0757	(714) 474-0131	(415) 962-0195
$3,000 - $7,000	$2,495	$185 (Also needs Hercules graphic card, $499)

For more information

Computer Graphics World
Publishing Company, INC.,
P.O. Box 122
Tulsa, OK 74101
(800) 331-5959

Publish
Subscription Department
P.O. Box 51966
Boulder, Colorado 80321-1966
(800) 222-2990

InfoWorld
P.O. Box 1018
Southeastern, PA 19398
(800) 544-3712

MacWorld
Subscription Department
P.O. Box 51966
Boulder, Colorado 80321-1966
(800) 222-2990

The Macazine
ICON CONCEPTS CORP.
P.O. Box 1936
Athens, Texas 75751
(800) 624-2346

Byte
Byte Sudscriptions
POB 590
Martinsville, NJ 08836

References

Whitney, Patrick. "Information, Computers and Design." Design in the Information Environment. 1st ed. New York: Alfred A. Knopf, 1985.

Nelson, Roy Paul. The Design of Advertising. 5th ed. Dubuque, Iowa: Wm. C. Brown Publishers, 1985.

Thompson, Keith. "Pagemaker Remains Chief Page Composer." InfoWorld, June 9, 1986, pp. 39-40.

Ebbesen, Ebbe B. "Desktop Publishing." MACazine. April, 1986, pp. 16-27.

Cavuoto, James. "Sophisticated Page Makeup Gets Personal." Computer Graphics World. September, 1985, pp. 49-52.

Burns, Diane, Venit, S. "Revised Editions." Publish. (Macworld. July, 1986, pp. 28-33.)

TV TOWER
PAUL RUTKOVSKY

Assistant Professor, Department of Art, Florida State University
Tallahassee, Florida 32306
Fellow, Center for Advanced Visual Studies, MIT
Cambridge, MA 02139

Fig. 1 TV TOWER, Paul Rutkovsky, Elizabeth Park, Hartford, CT., 1981

TH0164-4/86/0000/0045$01.00©1986 IEEE

The TV TOWER began as a simple video monitor on top of a crudely constructed tower. With as few words as possible, I will visually trace the tower's evolution into a major project incorporating a series of TV TOWERS with a main computer control system to be installed in Crandon Park, Key Biscayne, Florida.

The TV TOWER originated in the fall of 1980 when I first installed a mini-version in collaboration with the Music School at Silliman College, Yale University, New Haven, CT. The primary purpose of the work was for it to interact with the audience, on a visual and aural level, allowing them to talk and respond to the images on the monitor as other events (music related) were simultaneously taking place. The wooden tower was small, about 13-14 feet tall, because it had to be indoors, and at the time I was thinking only that it was a prototype for something else; it was unclear to me then what the TV TOWER would eventually become.

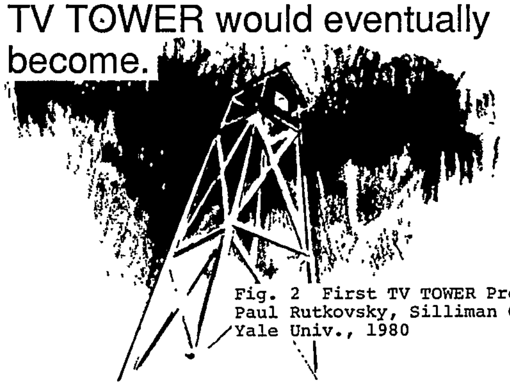


Fig. 2 First TV TOWER Prototype
Paul Rutkovsky, Silliman College
Yale Univ., 1980

Fig. 3 TV TOWER, Paul Rutkovsky, Elizabeth Park
Hartford, CT., 1981

In the summer of 1981 I was invited to submit a proposal for a two day outdoor installation in Elizabeth Park, Hartford, CT. I proposed to set up a 30-foot telescoping tower with a color monitor on top and a complete video/audio playback system (to be hidden somewhere in the shrubs). The playback system allowed me to talk live to people in the park without them observing my presence. It was difficult to convince the Park Commission that it was a viable project, but they finally agreed, with reservations. The installation was a complete success. Many people in the park, especially children, talked to the monitor and asked for directions to the rest rooms, refreshment stand, and other points of interest. The shock for many adults was the realization that an anonymous television in a park, on top of a pole, was specifically talking to them.

In collaboration with the Center for Advanced Visual Studies at MIT, I installed a third tower at Harvard Square, Holyoke Center, in the summer of 1983. This TV TOWER was similar to the Hartford tower, with the basic difference being the enclosed and populated environment at Holyoke Center. The audience seemed to be hostile and didn't appreciate the intrusion of an interactive television looking down on them. A lawyer threatened to sue me if I didn't remove all the power cords. This turned out to be an important lesson. I realized I required more control over the complete system, if I wanted to successfully involve large numbers of people with the tower.

I am now further refining the TV TOWERS with a design team in Crandon Park (35 acres), Key Biscayne, Florida. My objective is twofold: To provide a means of security for the park that is not "big brother looking at you" and to interface the towers with a computer that will control the monitor's swivel system (up-down, left-right) and video/aural imaging, while keeping in mind that everything should be hurricane proof. As a Fellow at CAVS, I will be working on the mechanical and electronic prototype for this project.

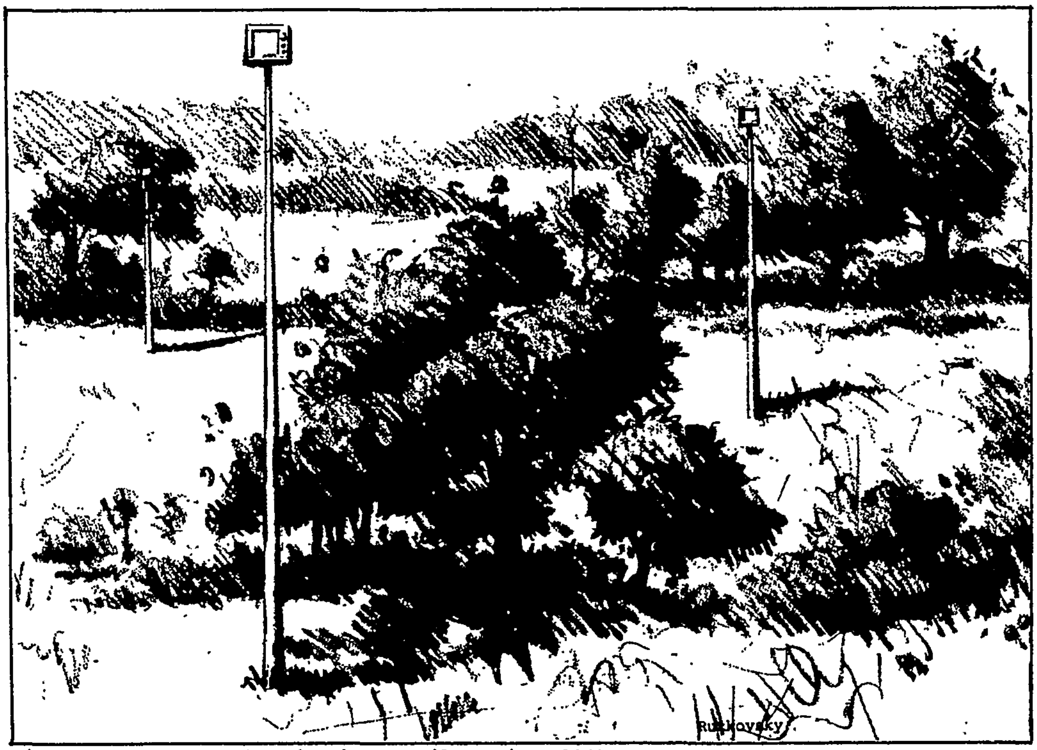


Fig. 4 TV TOWER, Paul Rutkovsky, Pencil Drawing, 1979

Trying to maintain the original humorous quality of the TV TOWER, with the whole complex park project and the design team of several other artists, has already become difficult at this early stage. However, it is exciting to work with scientists, engineers, and other artists on a major project that demands cooperation and requires all self-referential specialists to come out and share and communicate their knowledge.

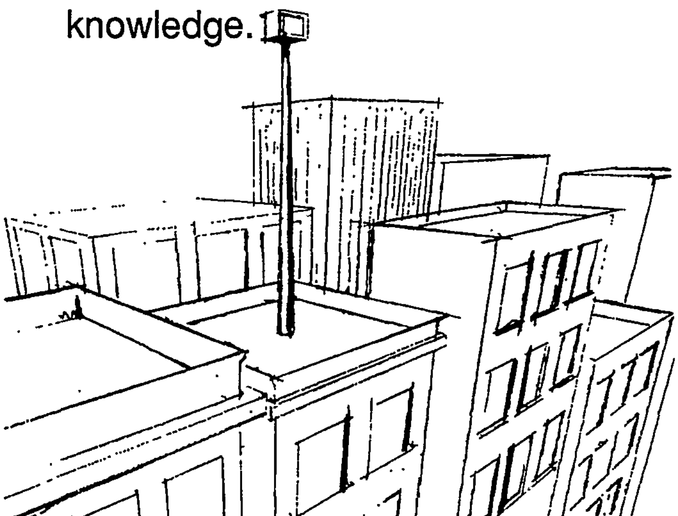


Fig. 5 URBAN TV TOWER, Paul Rutkovsky, Pencil, 1979

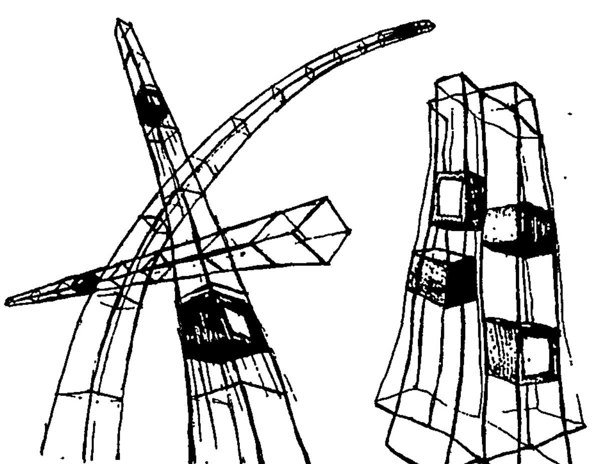


Fig. 6 MOVING MONITORS, Paul Rutkovsky, Pencil, 1979

Fig. 7 TV TOWER, Paul Rutkovsky
Holyoke Center, Harvard Square
Cambridge, MA., 1983

A COMPUTER BASED APPROACH TO SCULPTURE

J. Michael O'Rourke

New York Institute of Technology
Computer Graphics Laboratory
Old Westbury, N.Y. 11568

ABSTRACT

A computer based approach to the design of sculpture is described and illustrated with sculpture done by the author. The approach is shown to depend on the use of a wide variety of modeling and visualization techniques. The ways in which the approach simplifies or complicates the design process are discussed. Finally, some aesthetic and formal issues are discussed as they relate to and are affected by the computer aided design process.

1. INTRODUCTION

The motivations for attempting to use computer technology to assist in the design of sculpture are several. Computer modeling systems can potentially offer a great increase in design speed because of their ability to permit the designer to see an object before it has been physically fabricated. They also offer an enormous increase in the flexibility of making design changes for the same reason.[1]

For myself personally, there is a third motivation of an aesthetic nature. As a sculptor, I have an interest in certain formal issues which the computer, and specifically the system I have been developing, are particularly well-suited to handle. These issues and how they relate to the approach I will be describing will be discussed in the last section of this paper.

As attractive as the positive aspects of a *CAD* (computer aided design) approach to sculpture may be, the possible drawbacks of such an approach are potentially debilitating and must be overcome if the computer is to be more of a help than a hindrance. One major problem that must be faced is the limitations that most computer modeling systems impose on the types of shapes one can readily design. Another is the unsatisfactory nature, both in terms of quality and in terms of time, of the visual feedback the artist finds himself dealing with. And a third potential problem is the difficulty of getting the three dimensional data of the final design out of the computer and into the physical world. Unless a sculpture design system can overcome these problems, it will probably be more trouble than it is worth.

The system I have been using has gotten, after much work, to a point where it genuinely is more of a help than a hindrance for what I want to do. It remains, unfortunately, an approach which requires much more software and much more hardware than an individual artist can possibly afford. As with all computer technology, however, it is reasonable to assume that it (or something like it) *will* become accessible to the sculptor on the street, and that it will do so much sooner than we would guess would be possible. In the meantime, there do exist less powerful, but more affordable, *CAD* systems that an individual artist might find interesting.

2. DESIGN

The *CAD* approach to sculpture which I have been developing* and using for the last several years has been intended to answer several needs. First, it is intended to take advantage of the three dimensional modeling capabilities of computer graphics. Second, it should take advantage of the picture making capabilities of three dimensional graphics. Third, it is intended to be fast enough and flexible enough to permit creative

* This phrasing is not meant to imply that I have worked alone on this system. The entire support staff of our lab has been involved in this work and should receive credit. In particular, the major program components were developed by Pat Hanrahan, Paul Heckbert, Robert McDermott, John Schlag, Garland Stern, Jacques Stroweis, and myself. The piecing together of these components was done by myself.

spontaneity. And fourth, it must provide an effective means of translating the digital model into a real-world physical model.

In what follows, I will draw all my illustrations of the steps of the process from the development of a single piece of my own sculpture. By tracing the development of this one composition from its first beginnings to the final model, it should become clear how each step contributes to the whole process.

As with any other approach to sculpture, the first ideas are liable to be in the form of quick sketches. One sculptor might use clay for his/her sketches. Another might prefer wax. My own preference tends to be pencil drawings on paper. These are done quickly and often at odd moments—on a train, during a boring meeting. Beginning this way also works well for me because one aspect of what I am trying to deal with in my sculpture is precisely the relationship between a flat, two dimensional representation of a thing and an in-the-round, three dimensional representation of a thing. This issue will be returned to in the final section of this paper.

Like any sculptor's initial sketches, mine take into account both the technique I will be using to design the sculpture and the medium of the final piece. That is, one must know in advance and accomodate for both the strengths and the limitations of the techniques one will be using. In my case, my initial pencil sketches already take into account, for example, the fact that I will be working with flat surfaces and tubular curves. Another thing I know in advance is that both the compositional process and the fabrication process will be largely additive and constructive in nature (as opposed, for example, to the subtractive approach of clay modeling). The composition will consist of various pieces, shaped and then combined with other pieces. And finally, I know that the final piece is intended to be large in scale and fabricated of sheet metal.

Having gotten some initial idea of the composition, the next step is to begin modeling the individual pieces, or elements, of the whole. This is done with an interactive 3d modeling program** which runs on a real-time vector display device (specifically, an Evans & Sutherland MPS). Each component piece is designed and viewed as a wireframe image which can be manipulated in real-time by twisting a dial or a joystick.

The modeling program which I use is quite powerful and is extremely will designed in terms of its interaction with the user. It allows the modeler to design virtually any shape that can be represented as a set of flat, polygonal surfaces—whether a simple shape like a cube, or a complex undulating surface whose curvature is approximated by hundreds of tiny polygons.[2]

The sorts of shapes that can be modeled with this software include: geometric primitives, such as cubes, cylinders and spheres; extruded shapes, in which a two dimensional outline is drawn on a tablet and then pushed, or extruded, straight back into space to produce a block; surfaces of revolution, in which a two dimensional outline is rotated about an axis to produce the sort of shape one might make on a lathe; and lofted shapes, in which a series of two dimensional contours are positioned in space and then connected one to another.

Another modeling tool which I personally find extrememly useful is what are known as the "Boolean spatial set operations". These operations permit you to add and subtract objects to and from one another. A cross-shaped object, for example, can be modeled by adding together two cylinders which have been placed at right angles to each other. Similarly, a given piece can be gradually "shaved down" by repeatedly subtracting from it some other shape. In this case the second shape effectively serves as a chisel.

Figure 1 illustrates one shape as designed with this program. Note the see-through wireframe representation. This particular shape began as a two dimensional outline. This was then extruded to produce a block. Finally, one side of the block was given curvature by performing a Boolean subtraction from it with a large curved object designed for the purpose.

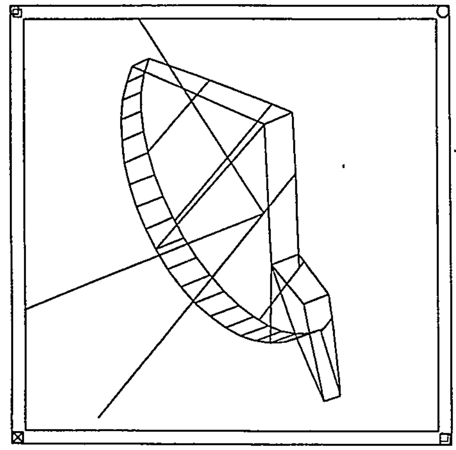


Figure 1. Wireframe representation of one component piece.

** This program and all the other software described here are NYIT proprietary.

Wireframe representations like the one here can be perceptually ambiguous. Consequently, our modeling system provides a mechanism for producing a shaded, hidden-surface raster image of the part being modeled. It is critical to the spontaneity of the design process that this sort of visualization capability be readily available. The longer the artist has to wait for visual feedback at any stage, the more constrained will be the flow of ideas.

Having modeled the various individual pieces to some satisfactory initial state, the next step is to begin combining them to form the whole composition. This is done on the same machine and again in wireframe representation. As with the modeling of the individual pieces, I can periodically pause to make a quick shaded image of the whole composition as it develops.

The primary advantage to combining the pieces in this digital way is that the three dimensional models we are dealing with have no physical mass and can therefore be made to pass through one another in the three dimensional space. Thus, a cylinder can be positioned to intersect another part, then repositioned to intersect it slightly less, then repositioned again not to intersect it at all—all without having to change the shape definitions of either the cylinder or the intersected part. In the physical world, each change would require re-modeling both parts. When the hidden-surface image of this is produced on a monitor we readily see what the physical intersections and resultant surfaces will look like.

As the composition of these flat polygonal surface pieces progresses, at some point I begin adding my other basic compositional element—tubular curves which move in and through the space of the composition. These three dimensional space curves are composed on another real-time interactive program, this one written by myself. Once again working in wireframe format, I read in the current version of the whole polygonal composition. I then use a 3-axis joystick to move a cursor through the three dimensional space of the program. This cursor allows me to draw a curve through space, reacting, as I do, to the polygonal composition which I have also placed in the space.

Having gotten a version of the tubular curve that I think I want, I once again can make hidden-surface images of the composition—this time including both the flat polygonal surface elements and the three dimensional space curve.

In addition to the real-time wireframe representation and the hidden-surface raster representation, there is another visualization technique which proves extremely useful. At any moment—from the very first modeling of the individual pieces to the final stages—I can also make hidden-line plotter drawings on paper. This has proved to be, for my personal working habits, absolutely critical to the success of the whole modeling approach. First of all, it permits me to make sketches which I can physically carry away with me. This releases me from dependency on the machines and vastly increases my flexibility. I no longer have to be in the laboratory to work on the piece. Secondly, the plotter drawings provide a link between the new computer techniques and all the traditional drawing techniques of the ages. In fact, these plotter drawings usually become rough sketches on top of which I hand-draw other ideas. Figure 2 is one such drawing. The basic plotter drawing has been fleshed out with some hand-drawn shading (so that I can see more clearly what is happening three dimensionally), and then has had some changes to the composition indicated.

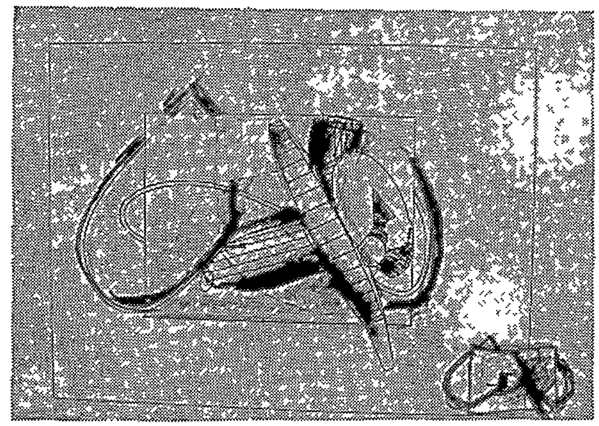


Figure 2. Plotter and hand-drawn sketch of the whole composition at an intermediate stage of design, front view.

All of these modeling and visualization steps take place repeatedly and in any order once I have started. For example, having studied some drawings, I might decide that one of my component pieces is not the right shape. I might then model a new piece in wireframe, combine it—again in wireframe—with the rest of the composition, look at a raster rendering of the whole thing, reposition the piece on the wireframe machine, look at several more raster renderings, and then make a set of plotter drawings to take home and mull over. Gradually, the composition begins to resolve itself, the changes I make become more subtle, and eventually I end up with a composition I like—i.e., which looks good from all angles and in all of the modes of representation available to me. Figure 3 illustrates one such final compostion as seen as a raster rendering.

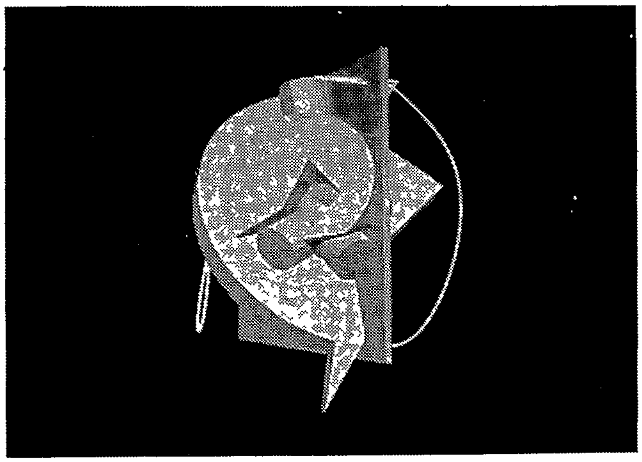


Figure 3. Hidden-surface raster rendering of the final digital composition, side view.

It has been my experience that most three dimensional computer modeling systems fail, from the artist's point of view, because of unsatisfactory visual feedback. It is not difficult for an artist to adjust to a system's limitations in terms of the types of shapes that can be made. David Smith, for example, made very powerful sculptures using only rectangular box shapes. But it is absolutely essential, if the flow of creative composition is not to be quashed, that the artist be able to quickly see what is happening visually. The visual information he gets must be complete enough for his purposes, and the effort required to get it must be minor enough that his creative thought processes are not interrupted.

Think of how an artist gets his visual feedback when working in traditional media. When working in clay, for example, he turns the model around in his hand. Immediate and full three dimensional visual information at the cost of (effectively) zero effort! This is tough competition for a *CAD* system to meet. But to the extent that a system does not meet it, the artist will be distracted from the real issues of composition by the technical problems of getting visual feedback.

The system I have been describing does not provide feedback as effortlessly or as completely as does working directly in clay. It does, however—and this was a principle goal of the system—provide enough visual information quickly enough and easily enough to be genuinely worth it, especially when its other modeling stengths are taken into account.

3. FABRICATION

In the Introduction it was mentioned that one of the major problems to be overcome by a computer aided approach to sculpture is that of translating from the digital data to a real-world physical model. We turn to that issue now.

Each final sculpture, as I have been using the system, is comprised of two major compositional elements—flat polygonal surfaces, and tubular space curves. The translation from digital to physical is handeld differently for each of these two categories.

The first step, once I have a composition I am satisfied with, is to use our modeling software to calculate exactly how each piece intersects and cuts any other pieces. Earlier we talked about various modeling techniques and mentioned the Boolean operators. These are used again here. If a cylinder passes through a cube, I can calculate exactly the shape of the hole which it will leave in the cube. I then have a new polygonal object—a cube with a cylindrical hole in it.

This procedure of subtracting piece *A* from piece *B* to produce piece *(B-A)* is followed for every situation in which two pieces intersect. Figure 4 shows one of the parts in our sculpture as it has been intersected and cut by every other part which touches it.

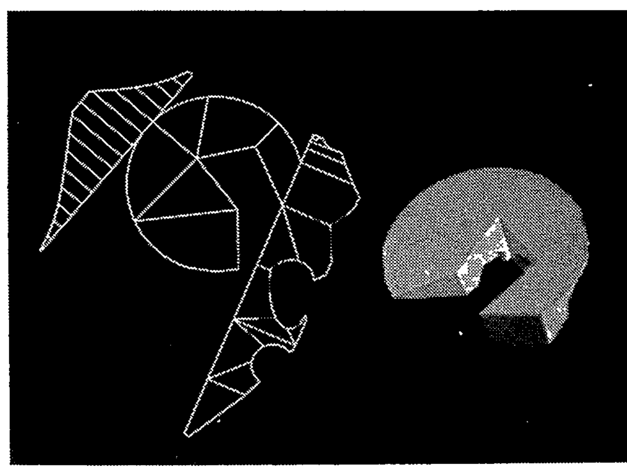


Figure 4. On the right, one polygonal piece with the appropriate cuts. On the left, the two dimensional flat "unwrapped" plan for the same piece.

Having gotten all the necessary intersections, the next step is to "unwrap", or flatten out each individual part. This is similar to what one might do with a cardboard box. A cardboard box is, like our pieces, a three dimensional object composed of flat, polygonal faces. If we remove all the staples in the box and flatten the cardboard out onto the floor, we will end up with a flat two dimensional pattern, or plan, for the object. This could then be folded back up into our original box.

The same approach is applied to each polygonal piece in our sculpture.[3] A program is used which allows me to indicate how I want each piece to be unwrapped. (There is more than one way to unwrap each object, and not all ways will be physically reassemblable.) The resulting two dimensional flat plan is then plotted onto paper. On the left of Figure 4 we see the two dimensional flat plan for the three dimensional object on the right.

When all of the polygonal parts have been correctly intersected, flattened and their 2d plans plotted onto paper, these 2d patterns are then cut out, folded up and taped together. (May the kindergartner in us never die.) Once all of the individual paper objects have been assembled, they are combined together into the whole polygonal composition, again with tape and glue. Everything fits together because each part has had scooped out of it whatever hole is cut in it by any other part.

Having assembled all the flat, polygonal shapes, it remains to create a physical rendition of the three dimensional tubular space curve. Unlike the procedure I use for the flat surfaces, the procedure used here is *not* a precise one-to-one translation from the digital to the physical. Instead, I simply use the hidden-line drawings of the different views of the composition as guides and "eyeball" the curve. For these first small paper maquettes, I use a heavy electrical wire for the curve and gradually bend and twist it until I have the shape I want.

For some sorts of tubular configurations, it is indeed possible to precisely reconstruct in a physical medium what you had designed digitally. Basically, one thinks of the three dimensional curve as consisting of a series of straight segments and calculates the angle of the cross-section between pairs of segments.[4] For the three dimensional curves I am dealing with however, the number of segements that would be necessary to give a good approximation of the complex and smooth curvature would be exorbitant. The benefit of precision that would be gained would be outweighed by the annoyance of having to keep track of a myriad of tiny tube segments. Since the point of this *CAD* approach to sculpture is not the *CAD per se* but the sculpture, a decision was made to go with what works best for the situation and to drop the computer technology at this point.

Figure 5 shows a complete, small-scale physical maquette, made of paper, cardboard, wire, glue and tape, of the sculptural composition we have been discussing.

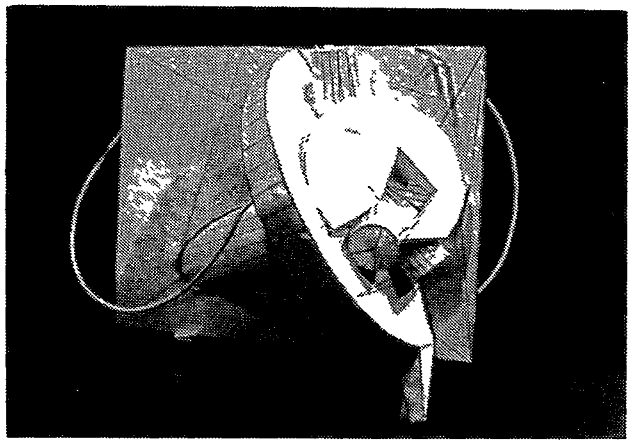


Figure 5. Paper, cardboard and wire model of the final composition. (11"x9"x6").

As of this writing I have not completed the last two fabrication steps for any of the sculptures I have been working on. The next step is to repeat, at a larger scale and including more surface detail, the paper/wire procedure, but using thin sheet aluminum and copper tubing. At this stage color will be added. The final sculpture is intended to be painted, with sharply delineated lines incised on the surfaces, just as they are drawn on the paper version. The final step in fabrication is to convince someone (?) to fund you and have the model built at its full scale out of industrial sheet metal and tubing. Final scale for the sculpture we have been using as our example is intended to be 7' high by 9' wide by 5' deep.

4. AESTHETICS

One level at which any tool, including a computer aided design system, can be useful is in terms of how it facilitates the task at hand. Is it easier, so to speak, to use chisel #5 or chisel #6 to knock off this hunk of stuff?

Another level at which a tool may be useful is in terms of how it might determine the nature of the task itself. If the task is sculptural composition, then the effect of the tool in this sense is in the realm of the formal and aesthetic issues that one attempts to deal with in the sculpture. It is these issues to which I turn now.

One of the concerns I have been working with in my aesthetics for a number of years is the interrelationship between certain contrasting, but hard to define, tendencies which I see in people. In my sculpture, these tendencies take the form of a counterpoint between contrasting visual elements—between straight lines and curved, between predictably geometric shapes and irregular, unpredictable shapes, between volume and space. An attempt is made to produce a composition in which the three dimensional relationships provide an analog to the conflicting emotional and psychological forces we experience within our selves.

The computer has been extremely helpful to me in this endeavor. One of its great strengths is its ability to quickly and thoroughly switch from one task to another. Its ability to do so, toggling, for example, from the generation of geometric primitives to the generation of free-form space curves, has enhanced my own ability to deal with these divergent types of visual elements. As a result, the compositions which I am producing now are more complex, more coherent and more subtle than they were when I began—or, I suspect, than they would be if I were not using this computer based approach.

Figure 6 is another such composition. It is from the same series of sculptures as the one which has served as our example thus far. Like it, it was composed with the computer approach I have described above.

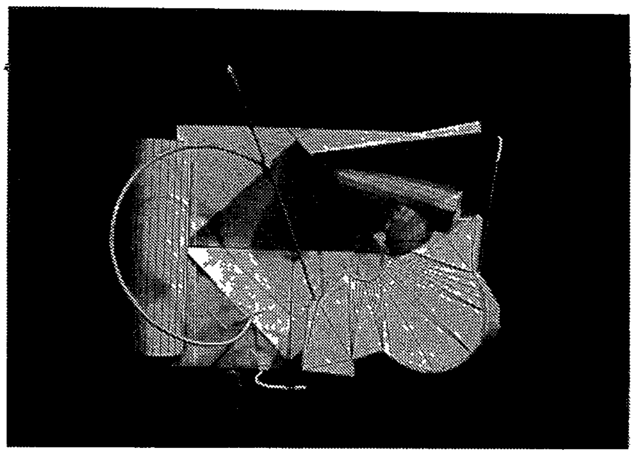


Figure 6. Another paper model from the same series of sculptures.

A second aesthetic issue which I have been working with in my sculpture and in which my *CAD* approach has been very helpful has to do with some questions of representation and semiotics. Largely as the result of my work with computer graphics, I have become very conscious of the inter-relationships between two dimensional representation and three dimensional representation.

Much of what I (and others working with computers) have been doing in the last years has involved two dimensional pictures of three dimensional "objects", with the objects sometimes being "real" and sometimes not. This raises all sorts of questions about the nature of representation, about perception, about symbols and about the nature of object-hood itself.

The sculpture I have described here attempts to deal in a very direct way with these questions. There is a sense in which it is, so to speak, sculpture about sculplture. The compositions consist, for example, of a series of flat two dimensional planes (the polygons) arranged into three dimensional configurations, with these built around a (nearly) flat "object", which is surrounded by both volumetric and (nearly) a-volumetric objects, some of which have purely two dimensional linear patterns incised on their surfaces, etc., etc.

All of this manipulation of formal relationships was inspired by and facilitated by my primary design tool, computer graphics. This is not to say that it couldn't have been done otherwise. Other artists, both sculptors and painters, have worked with similar issues in their art. For me, however, the approach which allows me to deal most fruitfully with these issues has proven to be based on computer technology. Perhaps it has lent its own special flavor to the way I think about things and to the nature of my imagery.

REFERENCES

[1] O'Rourke, M. "Computers, Sculpture and Three Dimensionality", Siggraph 1985, Course# 21 Tutorial Notes.

[2] Hanrahan, P., "The Winged Edge Geometry-Topology Modeler", NYIT internal technical document, 1984.

[3] McDermott, R.J, "Geometric Modelling in Computer Aided Design", Ph.D. thesis, Univ. of Utah, 1980.

[4] Smullin, F., "Pipedreams, A Complete CAD CAM System for Tubular Sculpture", *Proc. Symposium on Small Computers in the Arts*, 1983.

The Sculptor and the Computer
(An evolving view of a rapidly changing technology)

Jon B. Fordyce

Abstract

In this paper I will seek to introduce the computer as a sculptor's tool. I will offer a little background on current artistic applications of 3-D computer graphics and then describe some of features and costs involved with currently available hardware and software that might benefit a sculptor. I will discuss the relative merits of working with available software as opposed to programming your own application software. I will also describe how I became involved with the Computer. Finally, I will address the issue of how working with computer aided design can encourage creative growth in a sculptor's personal sculptural process.

A. Introduction

To date sculptors have done very little to explore the creative potentials of computers. There are numerous reasons including: lack of access, cost, limited capabilites for sculptural applications, and time required to learn a new medium. almost all of these blocks for sculptors are rapidly changing. Computers are becoming much more accessible because of a combination of falling costs for both computer and software, increasing personal computer 3-D graphic capability, and the coninuing development of 3-D software that offers improved capabilites and greater ease of use. The sculptor might use computers for 3-D designing or as a controlling mechanism in a sculpture. I will concentrate on computer aided design (CAD) in this article. Considering the electronic nature of the medium, the smell and tactile sensations of traditional materials will probably never be present, but you might agree that required efforts are justified as you consider the potential advantages of working with CAD. For the sculptor the primary advantage of CAD over <u>any</u> traditional single medium is that computers allow a sculptor to combine the advantages of perspective drawings, animation, the colors of painting, and the precision of traditional drafting techniques into one tool. Insert other program discs in the same computer and you are equipped for word processing, maintaining project accounts, and many other uses.

Let's say you have received a commission to create a large, painted, animated, sculptural fountain. Would you then value being able to create a solid sculpture model, try various colors and or pattern treatments on it's surfaces, be able to place it in an exact likeness of it's eventual permanent installation site, put the moving elements in motion as well as viewing the water patterns of the fountain heads, <u>and</u> generate precisely drafted drawings for the

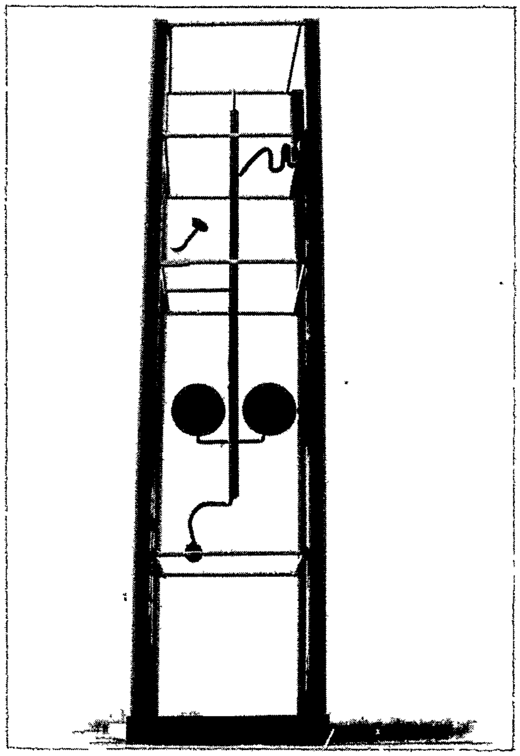


illustration 6.

<u>GRID 10</u>
(mild steel, enamel, vinyl)
1982
H.5' W.1' D.1'

patron, the architect, and the fabricator? All of those capabilites are available to the sculptor today. In order to reach the high level of capability (as suggested above), you will have to invest several thousand dollars in a CAD system, but with the purchase of a Texas Instruments 994A computer, additional computer memory, a disc drive, a compatible 3-D software package, and your color T.V., you can <u>begin</u> to get familiar with the medium for less than $500.00. If you can find a good price on a used TI 994A or a Commodore 64#17 in the classifieds, the price for a beginning 3-D CAD system might be considerably less.

TH0164-4/86/0000/0055$01.00©1986 IEEE

B. A little background

The use of computers for the drafting design applications of engineers and architects has been increasing steadily and dramatically in the past 15 years. Most of the commercial artistic applications have been for 3-D animation as we frequently see in T.V. advertisements, however increasing numbers of commercial design studios are acquiring sophisticated 2-D graphics software for many other applications as well. Largely due to the greater availabilty of 2-D paint software, an increasing number of fine art painters are also exploring the creative potential of the computer. A smaller group of sculptors, including myself, have been struggling to come to grips with this exciting new medium. The key question for the sculptor is: why should I invest the time, effort, and money to incorporate this new medium in my work? The answer(s) to that question are as individual as each of you and your sculpture work(s). Let's take a closer look at both advantages and limitations to computer aided design(CAD) for the sculptor.

Almost all of the CAD software on the market has been designed for industry and draftsmen in particular. Adapting this technology for the use of the sculptor is not always a clear path. The incredible diversity of mediums, styles, and techniques of individual sculptors places CAD in a situation where it may seem like a tool looking for a purpose. Even in industry, CAD's function in the design process is only partially understood and utilized. An equally overwhelming variety of hardware and software for CAD usage complicates this new medium even further. CIM (computer integrated manufacturing) which was first developed 20 years ago by Lockheed, is the direction of much of the CAD research and development presently taking place within the computer graphic field. This evolving technology promises a designer the ability to create a solid 3-D model at the computer and then send the coordinates to robots that can totally fabricate the elements of a product, assemble it, package it, and load it on a truck for delivery. The push to more fully develop this technology promises the development of more 3-D capable computer graphic software for the IBM-PC group of computers as well as for mainframe and mini computer systems.

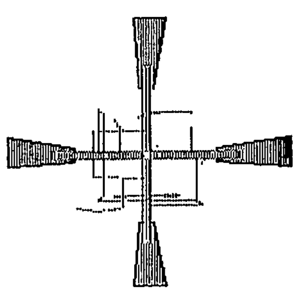
illustration 7.

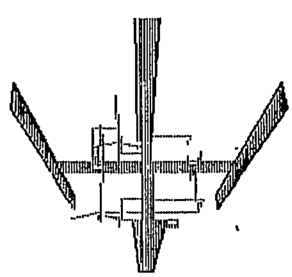
GRID 14
1983
Case Western Reserve University
Cleveland, Ohio

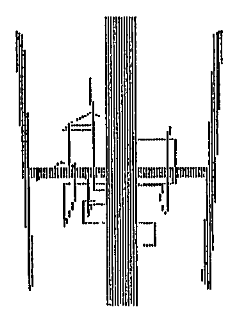
Computers are a vehicle to high technology change in seemingly every facet of society. Change usually meets resistance in regard to anything and CAD is no exception. There are complaints from some that: CAD systems are designed by computer progammers for computer programmers....not for draftsman or artists, and that most CAD systems cannot handle a number of unique problems. With all of this as background, let us consider some of the present capabilites of computer aided design.

illustration 8.

GRID 14
1982
(computer graphic image)







C. What is available for the sculptor?

The least expensive microcomputers (AppleII, Commodore, Atari 800, Texas Instruments 994A) have the most limited 3-D capable software. 4 colors of "Wireframe" or 3-D outlined forms that can be rotated are the "bottom line" limits of most software on the market for this memory size of computers, 48 - 128 kilobytes. Even with the computer memory size limitations of these small computers, you can create and store image forms on disc that can be called back as "building blocks" for the creation of new sculptural forms. For this group of microcomputers there are two 2-D drafting package on the market, to the best of my knowledge. POINTE #18 and CAD APPLE both were created for Apple computers (and the Apple clones) and offer many of the CAD features mentioned below. THE COMPLETE GRAPHICS SYSTEM#3 offers both 2-D and 3-D capabilites for Apple computers.

Artists, architects and engineers are increasingly using the IBM PC category of personal computers for 2-D and 3-D design work. "Wireframe" 3-D forms are also a limitation of most IBM PC (and compatible computers) basic CAD systems. Several of the upper end versions of current software packages allow for adding hundreds or even thousands of colors, solid modeling, "wrapping" 2-D patterns or designs around 3-D forms, antialiasing (smooth as opposed to "jagged" lines), "real time" movement, greater ease of access to stored file image forms (cubes, spheres, and the like), and greater speed of drawing. The increased capabilites over smaller computers are possible with PCs because any feature of computer software requires increasing amounts of computer memory. IBM PC's (and compatibles) presently come with a basic 256k and can be built up to 640k(or even higher with the addition of a hard disc drive or seperate coprocessor. Several software packages for this range of computers (Cubicomp modelmaker, Anvil 4000) allow for electronic representation of virtually anything the mind can imagine. The catch is money for most artists as these high level CAD systems might cost upwards to $30,000.00. Entry level CAD (drafting) software for personal computers (PCs) can be purchased for a list price of $49.00 #7 to $395.00 #12. There is a monochrome 3-D solid modeling software package for the Apple Mac Intosh for only $99.00#15. Generic Software#7 plans to introduce 3-D CAD software in the fall of 1986 that may make a giant leap toward affordable and easy to use yet highly sophisticated software for the sculptor.

D. How much will it cost?

It is almost impossible to state specific prices for computer hardware or software as there are so many variables, including complete "turn key" (everything you will need) package discounts, and sales. Generally, you might plan on spending $3600.00 to $10,000.00 for a complete IBM-PC (or PC "clone") "turn key" system which typically includes: computer, color monitor, graphics tablet, software, and a plotter printer). Depending on features you require, prices might be considerably lower or higher. If money is no problem for you then learning how to use CAD might be the problem, though most of the software manufacturers do offer training and phone call support. An increasing number of colleges and universities also offer CAD courses.

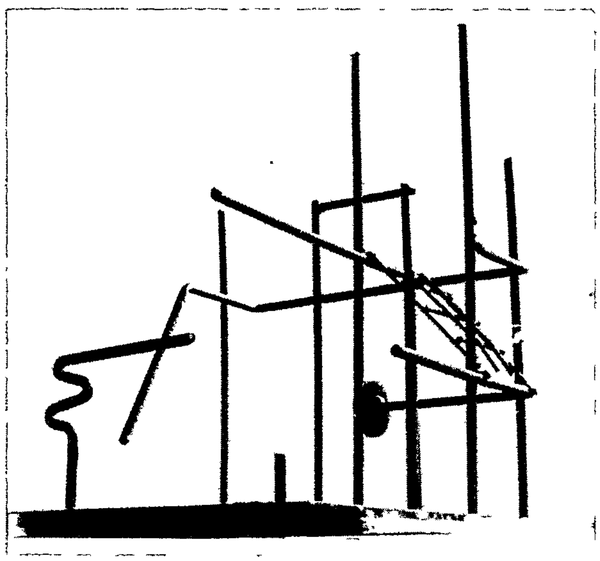
E. How I got involved with computers

In 1981 I was invited to create a monumental sculpture for Case Western Reseve University in Cleveland, Ohio. I decided to set aside my previously established styles and mediums and create something entirely new. I was not only flattered at the honor, but I felt a great repondsibility to find new concepts and materials that would chronical our times. After much consideration, I decided that the computer and the nuclear bomb have the most influence on late 20th century society. The bomb, however, evokes fear in my mind rather than the wonder that is generated in me by computers. I determined that it was time that I learned as much as I could about computers and simultaneously to begin to create sculptures that reflected my growing knowledge.

I began my research by taking an introductory computer course at a local college. The very simple discovery that programs and software make computers do something lead me to an equally simple and basic visual analogy that is consistent through what I titled the GRID SERIES. Geometric forms would symbolize the physical computer and organic curves would symbolize the human input. I then started evolving sculpture maquettes that seemed like designs that could be designed with computer aided design (though at that point I had not seen any computer generated art, I had read how 3-D computer graphics were based on the cartesian coordinates). These maquettes were made of forged and welded mild steel.

illustration 1.

GRID 2
1981
(proposed for corten steel)



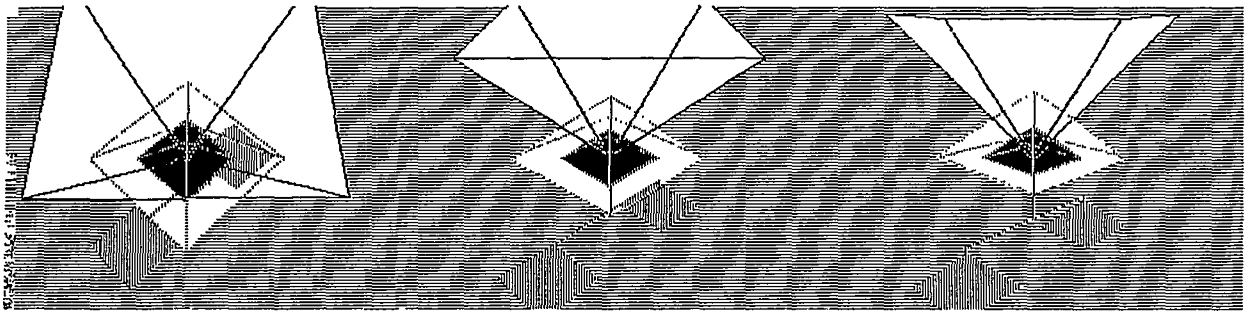


illustration 2.

PYRMD. 1
1982-4
(computer graphic image)

I continued my research into the possibilitys of CAD by visiting Skidmore, Owings, and Merrill in Chicago. In the early spring of 1982 SOM had the latest CAD capabilites for both 2-D drafting as well 3-D solid modeling, I learned more when I visited the home of a friend who was designing his architectural thesis project with the aide of 3-D animated graphic software for his microcomputer.

I decided that day that despite my right brained and mystical tendencys, if an architect can master this new mathematical 3-D medium, so can I. Because of the complexity of not only familiarizing myself with the new mediums of computers and stainless steel, plus the problem of creating forms that somehow symbolically says"COMPUTER", I felt a need to create an entire series that made an enduring statement about our late 20th century high tech enviornment.

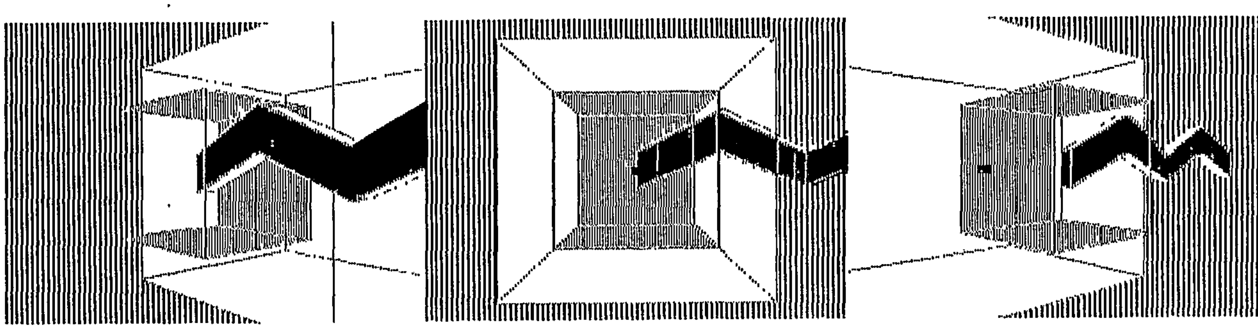
Over a period of two years, I created 21 maquettes (15 tangible and 6 "virtual" #2 - #4 designs) from which the University could choose. Some of these maquettes were intended for enlarged realization in polychromed mild steel, others in stainless steel, some in Corten steel #1, and yet others in a combination of those materials. Two pieces of the series were intended as end products as they stood #5 - #6. Case Western Reserve decided that they wanted GRID 14 #7 - #8 in stainless steel. In creating the maquettes for the series, I made two major discoveries: 1. Combining the inherently geometric quailtys of CAD designs with the equally inherent organic nature of forging was not easy. 2. I was fascinated with the idea of combining the ancient craft of forging with the leading edge medium of the computer. Somehow reaching simultaneously deep into our collective past and into an unknown future. Trying to find a way for the two mediums to work together in a new synthesis continues to challenge me. The path is not a clear one, but I like that......struggling to create with both materials and ideas....art that "breaks new ground" and speaks of the exciting times that are ours in a computer aided second reniassance."

Fortunately, there has been a continuing explosion of new CAD software since 1981. Indeed I discover something new on the market almost weekly in the computer trade magazines. The software manufacturers are continuing to seek to create 3-D software that is easier to use. Because of the intense development of computer integrated manufacturing, the trend toward a wider range of easy to use 3-D capable software with broader applications will likely continue. At this moment there is a need for more affordable 3-D software for the IBM-PC (and compatibles) as well as the new Atari 520ST and Amiga. Forenstance, VERSACAD (for IBM-PC and compatibles) in its basic form costs $395.00, to upgrade that 2-D basic drafting software to the point that it also offers 3-D wireframe modeling with hidden line removal costs a total of $1,200.00.

I continue my work with the computer #9 #10, and yet this work has not taken over as my total creative focus. I like the traditional processes all by themselves and have also continued to create spontaneous, tangible works in forged stainless steel and carved hardwoods. In retrospect, much of the work I have produced in the past 21 years has been figurative. I have yet to evolve the process that will allow CAD to help me with organic and figurative work to the same degree it can with my abstract and geometric series.

illustration 3.

EXP. 1
1983
(computer graphic design)



I look forward to creating sculpture that would combine computer animated graphics, lasers, and robotics, with tangible cast bronze, forged stainless steel, and carved stone. The possible combinations of traditional sculptural mediums with computers is endless. I continue to struggle with the "correct" useage of the computer in my art, but increasingly feel that "correct" does not exist. The computer is just another tool for the creative fruits of my mind....and yours.

F. Working in Virtual space

It is difficult to explain how it feels to me to be working at a keyboard as opposed to a forge. There are some objective observations that will give you a better idea of how the process is different. The most obvious difference with the computer is the freedom from the constraints of gravity. With this tool there is no need to necessarily be concerned with physical balance or structural connections. This design freedom allows the imagination to literally soar for this sculptor. Creating such gravity defying designs may also be of value as inspiration for physical sculptures that may be derivatives of the original computer generated form. Additional utilization for such designs can be found in printing the forms as direct physical works of art.

As I have used CAD to date, I have had to think in precise solid geometry which is a radical change from simply and intuitively watching the physical material as it gradually takes form. Depending on how your mind operates, this may not seem like a difficult task, but for me such an approach to the design - creative process has been a significant growth process.....and struggle.

Actually designing sculptures for exacting physical construction might well be the most important goal for most sculptors. Such a design process can be accomplished with very inexpensive computer systems in terms of 3-D wire frame outlines. It is true that in order to create anti aliased, solid, 3-D models, an artist may have to invest $30,000.00, but one should not discount the possibilites with lower end CAD capabilites. Often it seems that people who work with computers consider computer memory with horse power in a car. True, a Corvette is faster and more capable than a Honda Civic, as is a mainframe faster and more capable than a Commodore 64. More is better is often the clear implication. In both cases however, either of the cars or the computers will get you to the same place. Sometimes we forget that the primary tool that the sculptor uses is his/her imagination. Even less sophisticated 3-D computer graphics systems offer enormous potential for either practical or fanciful designing....even if math and logic are not your strongest abilites.

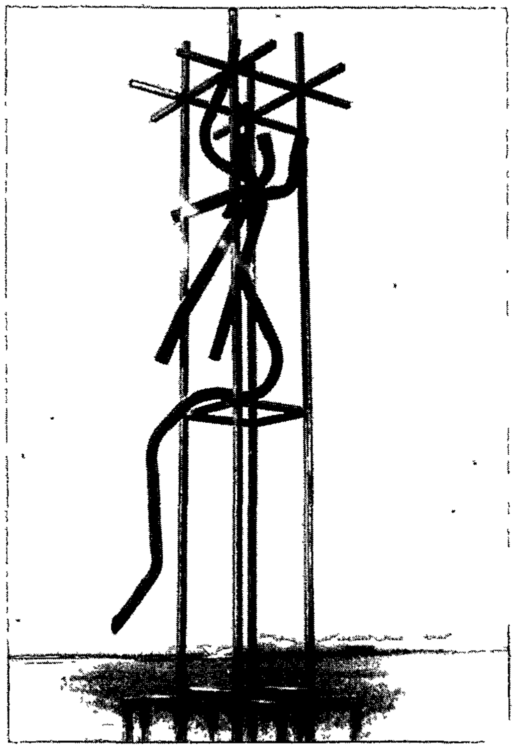


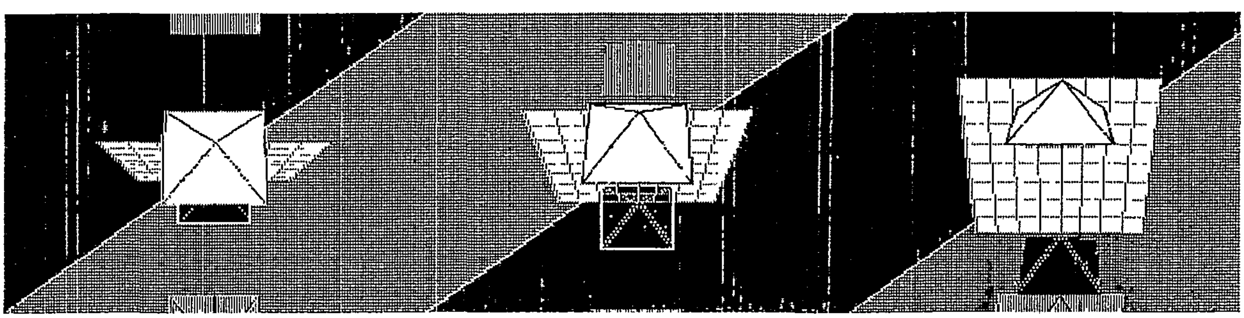
illustration 5.

GRID 11
(mild steel with enamel)
1982
H.8' W.3' D.3'

Time is also an element on at least two levels. One, When first working with this medium, it took a lot longer to design with CAD simply because I was just beginning to learn, now I can design 3-D forms at about the same or less time than with physical materials (depending on the physical materials and processes). Time is also an element in regards to how you see the form on the screen. Each different angle you see on the screen requires that the computer recalculate and redraw the virtual image at a new angle. With smaller CAD systems, as you rotate the form, these changing views are rather slow and not smoothly continuous in appearance. Larger 3-D CAD systems do the same things, but the movement appears smooth and in real time.

illustration 4.

DESIGN 3
1982-4



G. Programming versus software

The generally acknowledged first sculptor to use computers for 3-D design work, Frank Smullen, learned programming. Other artist - programmers continue to have the advantage to develop software for their own unique problem(s).

The potential disadvantage of programming for the sculptor is time. Learning to be proficient in the use of 3-D software developed by another can be very time consuming in itself, the additional time required to learn a programming language and then write a complex 3-D graphics program represents a <u>very</u> significant amount of time that one will not be producing tangible sculpture. However, if you desire to get involved with computer activated sculpture or interconnecting your tangible sculpture with electronic music or lights, then a degree of understanding how computer programming works might be a necessity. If the idea of programming appeals to you - do it. There is no substitute for an indepth knowledge of the technology you are using. For most sculptors though, purchased software will increasingly meet the needs of those who are primarily interested in final results.

Between learning programming and being limited to the capabilites of purchased software there is the possibility of "customizing" purchased software. Almost all drafting software allows the purchaser to create a "symbol library" of shapes and textures that a given sculptor may want to use repeatedly. These image files are saved on disc and called up as needed. You can do this "customizing" yourself or hire someone who specializes in such work.#2

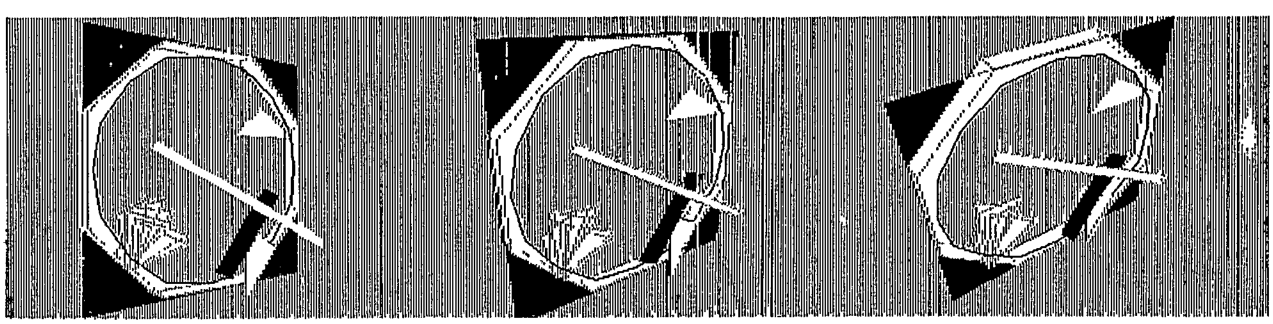
H. Intuition and Spontaniety

In my personal experience as a sculptor, intuition and spontaniety have been significant elements in my creative process. When I first used the computer I felt that I was limited and confined by my use of it. All CAD software allows only for <u>exact</u> designing. Allowance for spontaniety comes when you wish to rearrange previously designed elements or add new ones to your 3-D design. In this respect, CAD actually makes the spontaneous - intuitive aspect of the design process easier than with comparable changes on a tangible maquette. Spontaniety simply must be a bit more deliberate and the computer will yield very exacting results. As I continue to work with this technology, I find that I move with increasing ease from the keyboard to my studio. At the moment, I feel that 3-D CAD can be used in a variety of ways in my sculptural process. I have maquettes that presently exist only on computer disc, maquettes that exist only in stainless steel, and other works that have evolved as an interactive creative process. In other words, CAD has become an additional (but not all encompassing) tool in my work as a sculptor. I also have discovered that a spontaneously and intuitively created tangible model can be refined to exact proportions on the computer or a precise electronic (virtual) form can be given the qualities of spontaniety while actually making it.

illustration 9.

<u>GRID 17</u>
(stainless steel)
1986
H.22" W.19" D2.5"

illustration 10.



I. Summary

Computers are not about to replace stone carving, bronze casting, or any other sculptural process. Rather, with increasing amounts of competition in the computer market place with resultant greater availability, flexiblity, ease of use, <u>and</u> falling prices, CAD offers design process advantages to the sculptor working in traditional mediums as well as to those who wish to be on "the cutting edge" through the incorporation of computers in multi media works. Computers replace nothing, they add to our collective creative potential.

Time, expense, lack of tactile immediacy, a difficult to overcome geometric quality to CAD generated designs, and possible "in process" spontaniety constraints lead the list of the potential disadvantges for the sculptor. These are, of course, <u>relative</u> disadvantages and spontaniety (for instance) can still take place - either at the computer or in the physical sculptural process. Costs in money and time can vary widely. Depending on the nature of your work, computers may indeed be of no value to you. On the other hand, CAD can and will become an increasingly valuable tool for many. As more sculptors begin to incorporate computers in their work, not only will new uses be discovered, but the definition of art will likely expand through new collaborations of art with science.

The constraints of type space in this article, the diversity of both hardware and software, and constant changes in what is available have limited what I can share with you. If you seek further information, please feel free to contact me. If I don't know the answer, perhaps I can direct you to another source.

Jon Fordyce
301 W. Jefferson St.
New Carlisle
Ohio 45344

REFERENCE AND RESOURCES:

1. AUTOCAD, Boulton, Sandra, manager, Public Relations, Autodesk Inc., 2320 Marinship Way, Sausalito, Calif. 94703, phone (415)332-2344 (versions from $79.00)

2. AUTOCAD cutomizers, Chase Systems, 5518 Moccasin, Westerville, Ohio, 43081 ph. 1-800-235-6646 (source for customized versions of <u>AUTOCAD</u>)

3. <u>Complete Graphics System</u> (for Apple Computers) Penguin Software, P.O. Box 311, Geneva, Il.60134, ph. (312)232-1984

4. Cubicomp Corp., 3165 Adeline St., Berkley, Calif. 94703, ph. (415)540-5733

5. Digital Research Inc., (408)649-3896 (developing graphic software for the Amiga)

6. EASY 3-D (for Apple Mac Intosh), Enabling Technologies, Inc., 125 Armstrong Road, Des Plaines, Ill. 60018 ph. (312) 427-0408

7. <u>FAMILY COMPUTING</u>, "Preview of the Atari 520ST", Oct. 1985

8. Generic CADD, Generic Software, 13250 N.E. 40th St., Bellvue, Wash. 98005, ph.(800) 228-3601 $99.95 (for IBM PC and compatibles)

9. Halfhill, Tom R., "AMIGA and ATARI 520 ST reviews, <u>COMPUTE</u>, Sept. and Oct. 1985

10.ITT Information Systems, 2350 Qume Drive, San Jose, California, 95131, (800)528-1400

11. Leading Edge Products Inc. 225 Turnpike St., Canton, Mass. 02021 ph.1-(800)- 343-6833

12. <u>POINTE</u>, New Concepts Corp.,1161 N. El Dorado Place, Suite343, Tucson Arizona, 85715, ph.(602) 722-2245

13. <u>PRODESIGN II</u>, American Small Business Computers Inc. 118 South Mill Street, Pryor, Ok. 74361 (918) 825 4844

14. Raker D., H. Rice, "Getting CAD into more hands", <u>MANUFACTURING SYSTEMS</u> magazine, April 1985

15. <u>VERSACAD</u>, T&W SYSTEMS,INC., 7372 Prince Drive, Suite 106, Hunington Beach, CA 92647, (714) 847-9960

16. Walton, Roberta, "CAD-sculpture is new trend for atrium art" <u>CONTRACT</u> magazine, Jan. 1984

17. Wohler, Terry T., "The Phenomenal Promise of Micro-based CAD" <u>COMPUTER GRAPHICS WORLD</u>, Feb. 1985 PennWell Publishing Company

18. <u>3D-WORLD</u> by Amerisoft International, Texas Instruments 99/4A microcomputers, and <u>3-D CAD</u> software for Commodore 64 & 128 computers ($44.95), available through TENEX COMPUTER EXPRESS, P.O.Box 6578, South Bend, Ind. 46660 (219) 259-7051

NEURAL: An Artist's Book

George K. Shortess

Department of Psychology CU #17, Lehigh University
Bethlehem, PA 18015

Abstract

This article describes one approach to developing an artist's book using a microcomputer system. Emphasis is placed on the unique capabilities of microcomputers to generate variable formats and implement artistic ideas that are not easily generated in other ways. A copy of the book is included to illustrate the approach.

Introduction

As implied by the name, artists' books are books published by visual artists who are usually involved in all stages of the publication process (1). Many artists' books are one of a kind or limited editions. Others are issued in larger editions, typically using offset printing, although rarely aimed at the usual commercial market. This may account, in part, for the lack of general public acceptance of artists' books as a legitimate artistic medium. The content of artists' books varies considerably but the artist is usually making an artistic statement of some kind, e.g. autobiographical, social, political or formal. Often the page layout is uniquely suited to the statement, with words and pictures intermingled to form images with multiple layers of meaning. In general, artists' books are artistic statements which use a sequential page-like format as the medium for expression.

As I have indicated previously (3), this kind of alternate art form is quite compatible with microcomputer usage. At last year's Symposium, Joan Shafran (2) outlined the history of composition, showing that most, if not all, cultural groups that used the printed page experimented with visual variations of page composition. Using examples from Visual Poetry, she emphasized the blurring of the boundary between the traditional linear written word and the visual structure of the page, while pointing to the exciting possibilities of computer based compositional systems. The current explosion in desktop publishing offers a relatively inexpensive and flexible form for publishing a wide variety of book materials. As with other computer uses, techniques that were initially developed using larger computers have become available on microcomputer systems. The compositional capabilities developed on these larger systems, coupled with the availability and flexibility of microcomputers, opens exciting opportunities for artistic publishing. However, the software support for desktop publishing has typically been geared to the publication of text and line drawings that would be formatted into conventional column and page configurations. While this is not necessary, of course, the weight of the commercial market clearly pushes the development in this direction. Since artists' books often require unique formatting that may

NEURALN E U R A LNE U RAL N EU R A L

OUT OF THE CHAOS OF YOUR NERVOUS SYSTEM COMES YOUR EXPERIENCE

A BOOK PRINTED BY A PROGRAM
WRITTEN
IN APPLESOFT BASIC
BY
GEORGE K. SHORTESS
WITH AN
ANDERSON JACOBSON PRINTER, MODEL 831

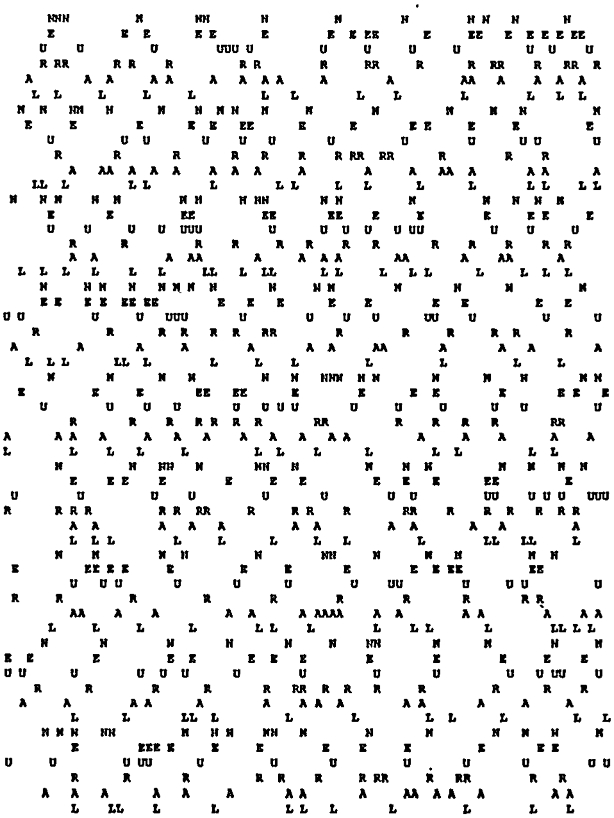
COPYRIGHT, 1986

PAGE 1. NEURAL

TH0164-4/86/0000/0063$01.00©1986 IEEE

not be done easily within these constraints, it is necessary to pay careful attention to the selection of general purpose software to allow maximum flexibility. The other option is to write a specific program to generate a particular book.

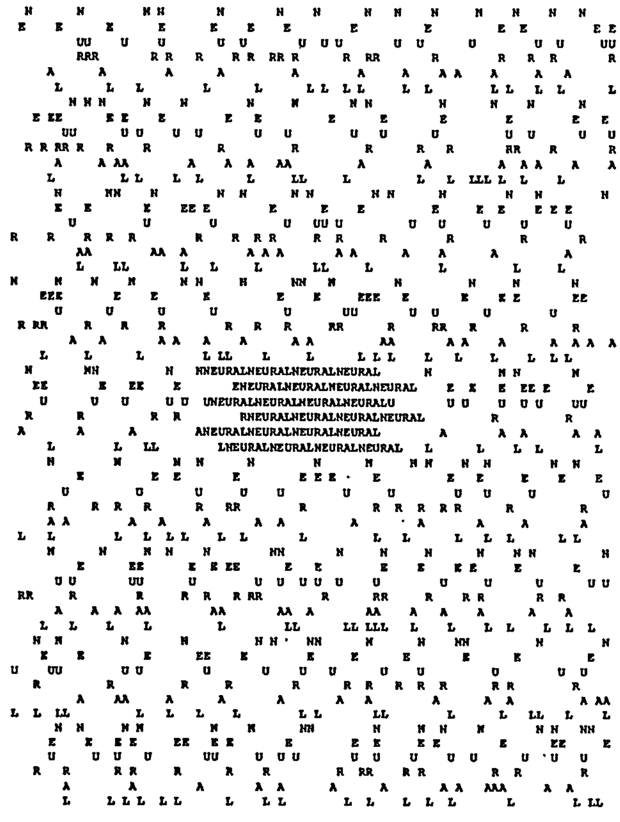
The book, Neural, that I am going to describe here was produced in this second way. Since I was not interested in a conventionally formatted edition, it seemed easier to write my own program. This, of course, could lead to writing more general software to create books



PAGE 2. NEURAL

of this kind, and I am working on the idea. In the meantime, I have reproduced one copy of the book in reduced form to show the result of this program. It represents the first complete product of a year or more of experimenting with a standard office system.

One other point should be made with regard to computer assisted art. There often seems to be the underlying assumption that the best computer art is state-of-the-art. I would argue strenuously against this position, and defend the idea that the quality of the artistic output is unrelated to state-of-the-art; the criterion for technology should be a function of the ideas that the artist is trying to



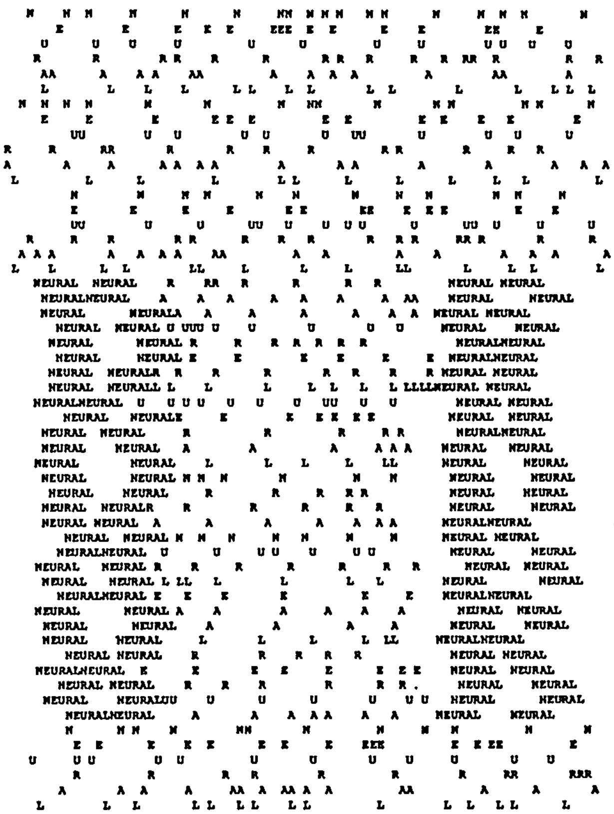
PAGE 3. NEURAL

express. Whether the artist uses oil paints or the latest high speed computer should be a function of the requirements of the artistic task. And just as oil paints may get in the way of expression so may a high speed computer. All this is by way of saying that my book project is not state-of-the-art, but given my goals, I feel that the technology was appropriate for the artistic statement that I wanted to make.

The Structure of the Book

As with all of my art work the ideas behind the book are derived from the metaphor of the perceiving nervous system. I identify certain formal features of the nervous system and use them as the basis of my work. The particular features are different for different works but the general process is the same.

There are two major neural characteristics on which the book is based. First, the nervous system is selective. Much irrelevant activity is generated by the constantly varying input from the sensory systems, yet we perceive in an organized and selective fashion. There is also some consistency in our perceptions, both within each system and across systems. For



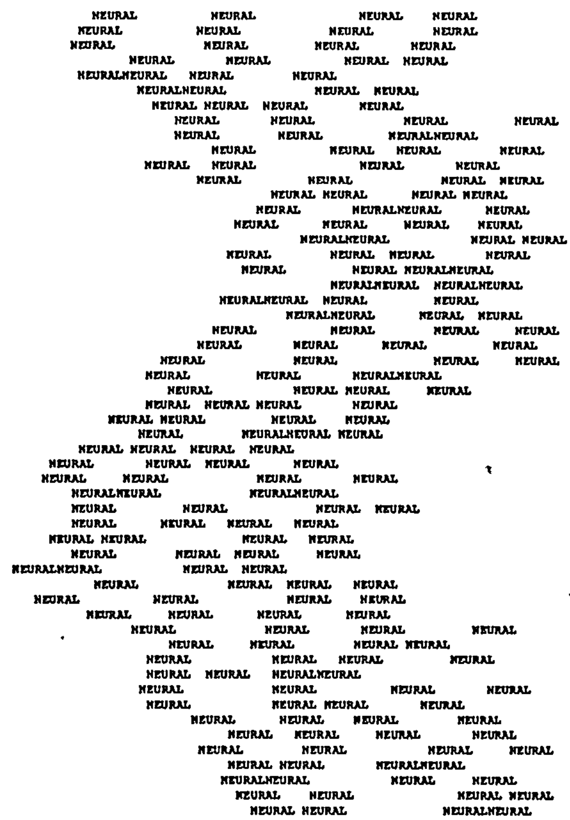
PAGE 4. NEURAL

example, we both will recognize the stimulus object who greets us on the street as the same friend.

The second characteristic, related to the first, is that while there is a degree of overall consistency, each of us is unique. Another way of saying this is that our nervous systems are structures within structures. We all have a similar overall form or macrostructure, e.g. a cortex, optic nerves, basal ganglia, a brain stem, etc. Yet because of a unique heredity and set of experiences each of us differs in the fine grain or the microstructure of this system, i.e. the particular set of connections and chemical concentrations that exist among the nerve cells. While the overall structural and functional similarities can account for the consistency between individuals, the microstructural differences can account for particular memories and modes of interaction associated with meeting the friend on the street.

How then can one express these ideas in a book? I have chosen to do it in a formal, structural correspondence between the ideas and the book. I have also tried to simplify in order to concentrate on these features. The subject of the book is neural, a word that I like and one that expresses the roots of the ideas in the nervous system. After the title page, only the letters from neural are used. Beyond that, the overlapping progression of pages is from the quasi-random spacing of the letters of neural in pages 2-4, to the quasi-random spacing of the word, neural, in pages 3-5. In pages 6 and 7 there is no randomness although page 6 looks more random than page 7 with its rigid vertical and horizontal format. This progression through the pages is a reflection of going from a degree of chaos to a degree of order, a process of organized selectivity. And this is true in all copies of the book.

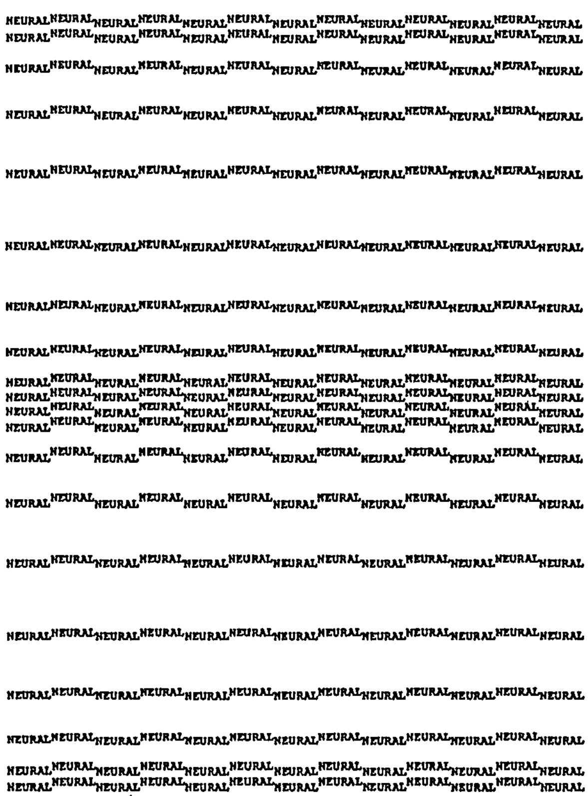
The individual structure within the overall structure is reflected in the fact that each copy is printed separately, in this case under direct software control (programmed in Basic). The sequence of quasi-random spacings of letters and words (pages 1-5) is different for each copy of the book. The computer program generating the text was separately run on the Apple IIe* for each copy of the book printed on the Anderson-Jacobson printer, Model 831. Each copy is then different in its microstructure while being the same in its



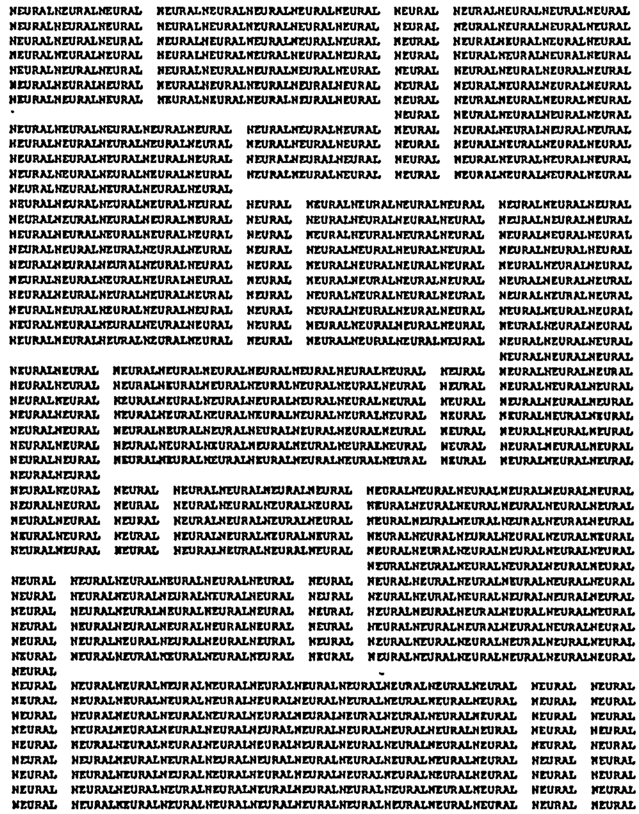
PAGE 5. NEURAL

macrostructure. It is random within constraints. A second aspect of this feature is that the viewer of the book can organize the letters of each copy on pages 2-4 differently depending on his/her own neural organization. She/he can see larger letters, basal ganglia or human forms in the overall arrangements.

Several other dimensions of the book should be mentioned. Each copy retains its sequential nature by remaining connected in the computer paper foldout form. I did this to show the computer origins of the book and because I have always been intrigued by this form of continuous book. Given these computer origins it is also interesting to think about where the book really exists. Because of the variations between copies it does not exist in any one copy. The way in which I prefer to think about it is that the book exists in the structure of the program used to generate the individual copies, in much the same way that our human perception exists in the overall structure of the way we are organized. This formal similarity is for me an essential part of the book's significance. The microcomputer has clarified these artistic ideas for me in ways that would be difficult using traditional media. Further development is needed in order to extend the boundaries of my art.



PAGE 6. NEURAL



PAGE 7. NEURAL

References

1. Lyons, J. (editor), Artists' Books: A Critical Anthology and Sourcebook. Rochester, NY:Visual Studies Workshop Press, 1985.

2. Shafran, J. "NEW PAGE: The Changing Face of Composition," Proceedings of the Fifth Symposium on Small Computers in the Arts, Philadelphia, PA, pp 91-96, 1985.

3. Shortess, G.K. "Art, Microcomputers and the Mona Lisa," Proceedings of the Fifth Symposium on Small Computers in the Arts, Philadelphia, PA, pp 13-17, 1985.

*Apple IIe is a trademark of the Apple Computer Company.

Hook Up: An Iconic, Real-Time Data-Flow Language for Entertainment

David Levitt
MIT Media Lab
Cambridge, MA

Abstract

We present a graphic programming language or kit in which black lines represent data equipotentials (wires) and icons represent functions and input/output devices. Music and sprite icons allow a user to learn the language by constructing many simple entertainment devices.

The language is briefly described here. One of the design goals of the language is rapid assimilation; if we have succeeded, this brief description and one sample program can give a good idea of what it does. The presentation will contain many figures, including music and 2-D 'cel' animation recording/playback; synchronized, interactive music and animation; and simple music composition algorithms.

The language runs on Macintosh computers. I designed the language with Burt Sloane in 1985. Implementation was by Sloane, myself, Mike Travers, Bosco So, and Ivan Cavero, of the Media Lab's Entertainment Group.

Hookup

Hookup is a real-time, interactive language intended for immediate experimentation and minimal documentation. To start the program the user clicks twice on the Hookup icon. A display as in Figure 1 appears on the screen.

To compute something, a diagram is connected with the wiring tool. To provide source data, an Input Box is dragged onto the screen (a cursor will blink) and a number is typed into it or a slider or button is connected to it. Information generally flows left to right. You can't connect two sources.

To see what value is on a wire post, it is connected to an Output Box.

When you want to connect something with the wiring tool, it is moved to the place that you want. The user checks that the connection turns black (highlighted) before release of the mouse button. The wire posts can be used to make a neater diagram. You can move icons and posts around with the dragging tool. You

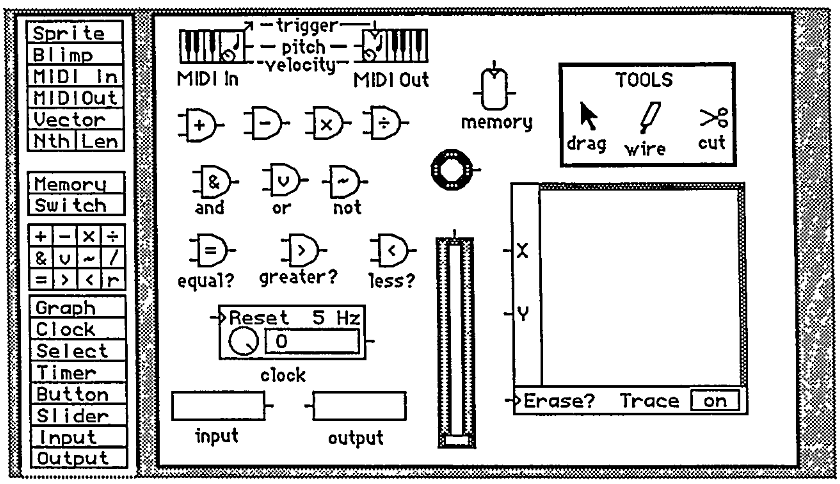


Figure 1

TH0164-4/86/0000/0067$01.00©1986 IEEE

can also cut icons and wires with the scissors, or just drag them off the edge of the main screen.

The following logic operations are supported:

- and &
- or v
- not ~
- equal? =
- greater ? >
- less ? <

They use a zero for true and a non-zero value for false.

The button sends a one (1) when you are pressing it, and sends zero (0) otherwise. The slider sends a number between 0 and 99.

The triangular mark on the Memory, clock and MIDI Out, icons means that inputs respond to false -> true transitions. For example, in Figure 2, if you push the button you will store into the memory cell whatever value is connected on its left side, reset the clock and send a note of the given pitch and loudness to the MIDI music synthesizer.

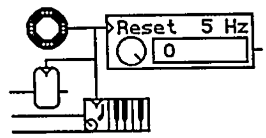


Figure 2

Now, every time the button is pressed, a new MIDI event will be played via the MIDI Out icon, and the data will shift right one. The loudness can be controlled by adjusting the slider which has been connected to the MIDI velocity input.

We can make an oscillator and connect it in place of the button as seen in Figure 3. When you press the button OR if the clock output equals 2, the clock is instantly reset and counts from zero again. After the button is pressed the clock output will simply oscillate between zero and one.

In this figure, we have also dragged out another *MIDI* icon to send NoteOn events with velocity of zero to turn notes off. (This is often helpful with the CZ-101 and Yamaha DX instruments.) You can see that we tapped separate cells in the shift register, 1, so releases for an event are delayed with respect to attacks.

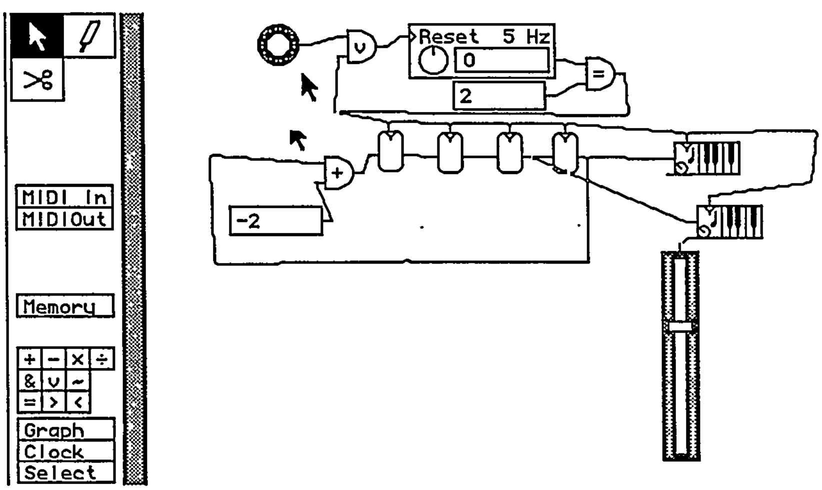


Figure 3

The most confusing bug you may encounter is that sometimes typeout in Output boxes does not update when it should; you may have to touch the output boxes' rectangle with the mouse to see their current values.

Figure 4 shows a quick sketch of a musical canon generator to help people get started. If you play in notes from the MIDI keyboard, four pitch events will get stored. Right now the MIDI In icon only notices NoteOn events, so if you have an instrument like a Kurzweill 250 that sends NoteOff events at key releases, the shift register shown (built out of memory cells and wires) will store four pitches. If it is, for example, a Casio CZ-101 that sends NoteOns with velocity zero you will store the pitch of each attack *or* release.

Now, we can repeat the 4-events again and again, sending them back to the sythesizer using the MIDI Out icon.

In Figure 5, the MIDI input module has been disconnected (actually it should be left on the screen somewhere) and the trigger has been replaced with a button to push with the mouse. The shift register data output has been been looped back to its input. There is an additional trick in this diagram. A number has been added to the pitch datum before looping back. With -2 in the augend, this melody will descend by a major 2nd (2 half steps) every time it is played.

Short Cuts

Some construction short cuts are available:
First, select the dragging tool by touching it with the mouse. Then, hold down the Option key, this turns it into the wiring tool. If you hold down the Option and clover together this turns the tool into scissors.

When Auto Connect is selected (the default), bringing an input near an output will create a black highlight. Releasing the mouse while the highlight is on will connect the two.

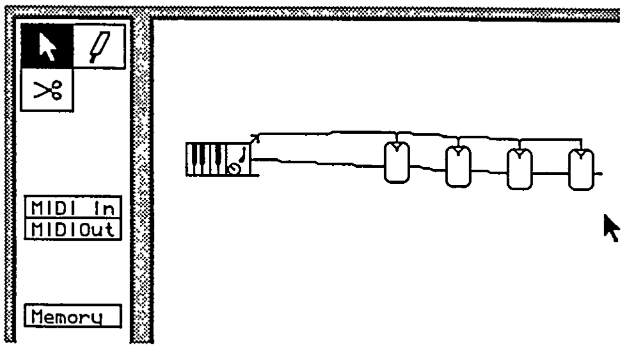


Figure 4

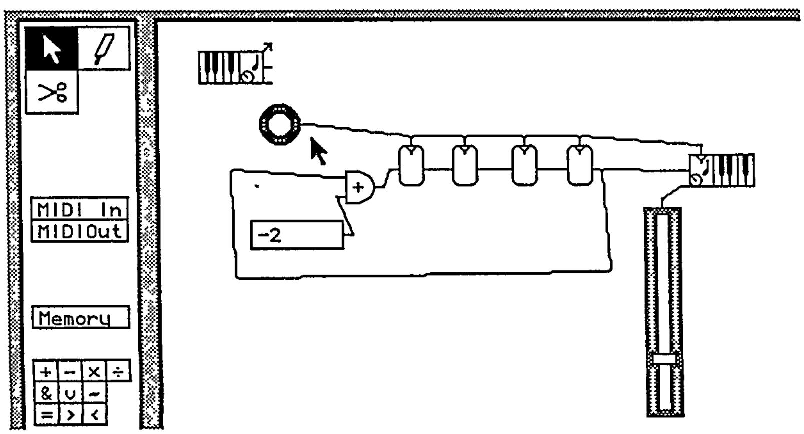


Figure 5

Summary of advanced icons and options:

Ratio --/ returns a rational number, unlike + which truncates to an integer.

Remainder--r produces a remainder, the sign is the same as that of the dividend (top) argument.

Switch--Routes either of two inputs to the output. Connect binary input to top or touch to toggle.

Memory--Dialog sets size and touch sensitivity. Touch trigger tick to lead input from left.

Vectors

Vector--new element goes in on the left; vector comes out on the right, length shows in display
+ means grow (add new element onto end)
0 means clear
- means rub out last element
If triggers appear simultaneously, only the highest priority operation is performed; the priority is: + 0 -

Len--takes a vector as input and returns its length.

Nth--retrieves the Nth element of a vector. The vector and index inputs can be swapped.

Mem(ber)--returns 1 if the value is an element of the vector. Inputs can be swapped as in Nth.

Real-Time Music and Animation

Timer--like the Clock but counts at 60Hz. The Timer's display updates several times a second.

MIDI In or Out--top is trigger (on new attack or release, middle is pitch, and bottom is velocity. The MIDI Out dialog (double click) lets you select channel number. If you set the radio button in the MIDI Out pallette dialog, the MIDI channel will be incremented when each copy is made.

Sprite--connections are (counterclockwise) vertical, horizontal and cast member integers. The Sprite dialog (double click) lets you select the channel (1 is back) and draw/erase modes. The Sprite pallette dialogue lets you change the cast member file. (Delete any existing Sprite icons first).

Mouse--continuously produces X (increasing to right), Y (increasing downward) and mouse button state.

Graph--continuously plots points at given X and Y coordinates. Erase is trigger input.

Blimp--connections are (counterclockwise) left, right, and vertical propeller directions (-1,0 or 1). Use this only if you really have a radio-controlled blimp on your Mac.
Only one MIDI In or Blimp icon can be dragged out because the system will not let you drag out icons that contend for the serial port resource.

Special menu

Auto-connect: Inputs will automatically attach to outputs if you release the icon while the connection is highlighted. Invalid connections will beep.

Hookup is a software kit developed at the MIT Media Lab c 1985, 1986 by Burt Sloane, David Levitt, Michael Travers, Bosco So and Ivan Cavero. Thanks to Alan Kay and Apple Computer for support. Video Works animation drivers wre provided by MacroMind. David Levitt may be reached at (617) 253-0608 or via electronic mail to levitt@media-lab.mit.edu.

The Art of Storytelling with a Computer

Rob Morris
V_Graph Inc

Clearly one of the oldest and most venerable of the arts is storytelling. This activity was used to carry on the oral traditions of the ancient peoples. Even today then the puppet master and his apprentices enter the villages of Bali to weave the latest version of the ancient Ramayana with light and shadow, virtue and vice, events both ancient and current. Even now the whole village turns out for the all night adventure into the imagination and soul of the people. They say that anyone who can see the play of the shadow puppets, or hear the cries and laughter is protected by the magic of the gods themselves.

One of the critical elements of storytelling is that the story has to be responsive to the listeners. It is helpful not to think of a story as a thing in an of itself because a story is what you hear, and what your mind builds for itself as the telling continues. A good story teller is one who can respond to his audience. Who can modify the telling to suit the mood and interests of the listeners. The story tellers of Bali do not just set up their puppet theater and grind through the old stories again and again like playing the VCR too many times. Instead they spend a while in each new village to pick up on the local gossip, and to learn the new political feeling in the village. When it becomes time to start the play they are well armed with the latest jokes, stories, and news about the current village, as well as all the villages they have visited in their trip. This information is incorporated into the ancient myths and sacred stories they tell. In this way news is spread, and the villages are brought together in the safe arena of the puppet theatre. This method of story telling reaches into people's lives and gives them the old and the new. It teaches and reaffirms as well as entertain.

The use of computers to tell stories is just now starting to become wide spread. With the advent of video games and interactive videodiscs technologically aware people started to see the possibilities of this new medium. Right now people are using these tools to teach, to show, and to sell. There are those who poo-poo these activities as not being art. Sure there are hacks, and there are poor teachers, but that has little to do with what can be done, and the realization of new ways to tell a story, or new kinds of stories that can be told.

The unique element to telling a story with a computer is that the experience can be interactive. There are a number of different methods for creating interaction between man and computer. I am not referring to hardware devices such as touch screens, graphics tablets etc. I am referring to ways to use the computer's power to enhance the experience for the person receiving the story. Interaction between man and computer can be incidental, or cumulative. It can be have a random elements or encourage intervention on the part of the listener.

There are many of computer programs out there that are being passed off as being "interactive" which are something else altogether. In its most degenerate form there is interruptive media. This poor facsimile for interactive media is characterized by the fact that the user is required to press a button, or the screen, in order to go to the next predetermined part of the show. In this case any button will do, and you can be sure there will be another stopping point waiting for the user just ahead.

TH0164-4/86/0000/0071$01.00©1986 IEEE

A more popular method of designing interactive media is the menu branching method. This is a simple to create and easy to understand way of making responsive presentations. Just ask the audience what they want and give it to them. What could be more direct, and less subtle? In many cases this is a sufficient method of allowing the audience to interface with the story. They can choose the next thing they see, and the next, and the next. The storyteller/designer can channel the audience into more and more intricate and obscure plots and subplots with this method, and a great time can be had by all. However the story will stop and wait for the audience to respond before continuing and this periodic halting can harm the flow of a good story.

Another approach can be found in some of the text or adventure games. Many people have had the opportunity of playing with one or another version of "ZORK". This game is a kind of branching structured game which allows the player to find his path in what seems to be a maze. (Technically this is known as a graph problem.) Without going too deeply into the internal structure of graphing algorithms let it suffice to say that it is possible to think of finding the goal of the game (transversing the graph) as finding your way along a map route by using your finger, and finally arriving. One of the interesting aspects of this kind of game is that the user is allowed to enter any word or sentence in response to the game prompt. Realistically it is only possible for a computer program to respond to a certain number of word combinations that the player might enter. The heart yearns for a larger dictionary and more sophisticated parser which will respond to any thing the player might enter. I'm afraid we will all have to wait for that to happen.

There are programs that do not ask you to make an entry, but instead default to a predetermined path if you do not. If you do make an entry the program will change its course accordingly. These programs are used to teach interventions skills, but this technique can be used to allow the audience to change the course of the story show in the middle of something if there is room for him to do so.

Because the computer is involved it is possible to record the pattern of responses that the audience is making, and to use that pattern as a indicator in choosing what will happen next when they do respond again. In this way you have the direction of the story being determined by the user in a more indirect fashion. The storyteller/designer regains some control, but uses it in such a way that acts as if the storyteller is now under standing the audience and responding with wisdom gained from the experience of the past responses, instead of blindly jumping every which way the audience may choose.

One interesting thing about all this is that it is possible to create simulated environments, to build models of the real world where each element in the story has its own rules and when the viewer makes different elements of the story interact in the simulated environment you can have interesting and unique results, even from the designer's point of view. This is where the interaction between the viewers and the program starts to create its own unique effects and set of occurrences. Now you really have something because the viewer is truly creating his own role in the story. The storyteller/designer has allowed the viewer into the story. He has limited the effects the viewer can have by defining the roles of all the other players. Let's take a simple example. I designed a program that would allow the viewer to construct his own computer network. By moving images of microcomputers, phones, printers, etc, the viewer could build his own system. Then he could go on to operate the system. He could make voice, data, or voice and data transmissions. People enjoyed this because they had the ability to build a system that I, as designer, could not have predesigned for them to choose from a menu. I have no idea how many thousands of combinations of system elements viewers could construct, but I do know that there are many more than I could have given them if I have used a different structure in my approach. The point to this is that computer power allows for this kind of flexibility in getting your story told. Now if I had characters instead of images of computers, phones, etc I could have given each of the characters attributes that would have defined their behavior in any allowable situation. For example "Bill is stingy". The viewer wants Bill to take Sue to lunch. He does, but he gets her to split the check at the end of the meal. His attribute of stinginess is carried through all his actions. You can give multiple attributes to people and give weights to each so it is possible to make things pretty complicated. The audience will appreciate this.

In any case there are a number of different techniques that can be employed in creating an interactive story. I have just described a few. Don't forget that they are just techniques, not an end in themselves. Interaction with a computer is not storytelling! It can just be pushing buttons to see what is going to happen next. The single most important aspect to this is the understanding the designer/storyteller has for his audience. He must be able to engage the interests of the listeners, and compel them to become part of the story. The hard part is that the designer/storyteller is usually not around to help out, and his work must stand on its own. This is what I refer to as "The computer program as performance art". In order to be responsive to many different kinds of audiences it is possible to have the first few interactions channel the viewer onto a track that is appropriate for that viewer's outlook. The track can be changed, but once it has been taken it is harder to get off it than to stay there and explore its attributes. If one story is going to reach the hearts of all people it is possible to structure it in such a way that different kinds of people can see the story from their own point of view. This means that the storyteller/designer must be able to think about his story in more than one way at at time. This is not easy. He must also be able to devise ways for the viewers to interject their own points so the story is carried forward without disrupting the flow of the story. Other aspects to this are the practical aspects of time and energy. Since it is not likely that someone will spend his whole life on a single story it behooves the storyteller/designer to find ways to reusing elements from one track in the other tracks. Now instead of having a branching type story where you keep going down unique lines in the story until one of the lines ends, you are back in the land of graphs, or thinking of the story as a kind of map, where it is possible to get anywhere from anywhere on the map if you only know the route. Each path must be consistent with the others and whatever chronology that is maintained in the story must be maintained regardless of which path is taken. Just to make the experience a little more interesting you may want to keep track of the viewer's responses and give him a look at his path based on his personal history in the story. In this way the viewer becomes part of the story, and it does matter which button he pushes. He is a player as much as the characters in the story are players. His actions not only effect the characters, but they also effect him, and his ability to do other things in the story.

Now that I have mentioned some of the main points of whatgoes into interactive storytelling I will explore some of the ways to create an interactive story.

In general there are two routes: the first is to program the computer in some programming language like Basic, Pascal, C or whatever your favorite is, the second method is to work with an Authoring package. An Authoring package is a software program that lets "non-programers" create interactive presentations. Authoring packages were developed for instructional designers, and videodisc producers who prefer to design, and not program. These packages let you devise a series of events which will be performed by the authoring software. Some are more powerful and flexible than others, and there are some 20 that are currently on the market. I will not review them, but remember that none of them has every thing you want. You might like to think of them as prototyping tools to be used on your way to creating an interactive story.

You are also going to need some kind of format in which to present your story. There are three popular formats to consider: computer text, graphics, and video.

Computer text is the easiest one to do, and can be presented on any general purpose computer. Unfortunately many people find reading computer screens to be tiresome. You can spruce up the look of your story by using one of the many windowing programs around and include color text. In many ways text allows the storyteller/ designer to have the most flexibility. This is so because the text can be organized in such a fashion that you allow blocks of text to be reused in different places. This reuse of text can stretch your resources and let the viewer to interact with the story in a way that has been described above. You can add and delete key words in the text according to the values of variables which get their value from the choices the viewer makes, and thus change the meaning, e.g. Marsha said "John I love (hate) you".

Computer graphics are a little harder to create and to work with in a programming environment, however it is possible to create a story at home with just your own talents and some specialized tools. I am not sure how many authoring packages there are out there for this, but you may want to look at PC Storyboard to get an idea what is on the market. These stories in computer graphics are fun to have around. There are like an ex tended video game, and with a little creativity they can be made to be interesting to people who are want more than a test in hand-eye coordination. Again, graphics allow the viewer to manipulate the story. Images can be used and reused to construct a variety of full pictures, and it is possible to put text on top of the graphics to carry the story along. One of the nice things about graphic representations is that they can be abstract and compelling at the same time. They are flexible enough to allow for on-screen manipulation of the image, and they can be shown on a wide variety of graphic compatible machines.

Videodisc presentations are quite glorious, but they are also quite expensive to create. You have to script, shoot, and edit the video tape. Then you have to send the tape out to have it turned into a disk. After this you have to program a computer to control access to the scenes on the disk. Of course you need an interface card of some kind, maybe a touch screen, and graphic overlay too, how about a voice activated system while you are at it with a motion detector. I would not recommend this route for the casual experimenter. However the results can be very pleasing. You do lose some of the on screen flexibility, but you gain much in dramatic realism.

In conclusion, remember that the methods and tools described above don't amount to a pile of paperclips unless you have a good story to tell. You can use all the bells and whistles you can find to amaze your friends, but technology will not carry a poor idea. So work on the basics of your tale before you complicate your life with a lot of fancy gizmos. Use the computer for what you need it for, not to wow yourself with your own cleverness but to bring your audience closer, and to let them feel your story for themselves. Learn how to reach their hearts, and the tools will come in to play as you need them.

 V_Graph is a system/ software house thatspecalizes in creating inter-active presentations for training, sales, and promotion.

AUTHOR INDEX

PROCEEDINGS

6th symposium on small computers in the arts

Library of Congress Number 86-45867
IEEE Catalog Number 86TH0164-4
ISBN 0-8186-0755-6

COMPUT
SOCIETY
PRESS