BYTE

THE SMALL SYSTEMS JOURNAL®

DECEMBER 1986 VOL. 11, NO. 13

$3.50 IN UNITED STATES
$4.25 IN CANADA / £1.75 IN U.K.
A McGRAW-HILL PUBLICATION
0360-5280

Graphics Algorithms

Graphics Algorithms

THE WORD GRAPHICS, derived from the Greek *graphikos*, to write, can be loosely defined as the process of representing a three-dimensional entity on a two-dimensional surface using mathematical rules of projection. The latest generation of personal computers, such as the Amiga, Atari, Macintosh, and advanced workstations, all emphasize the use of graphics. Business, engineering, medicine, and the arts have all reaped benefits from advances in the art of computer-generated images.

Programmers are continually searching for better methods, or algorithms, for representing these three-dimensional entities on computer displays. The quest for more speed, animation, and better color and realism in computer-generated graphics continues unabated.

Our December Theme articles present some interesting and innovative algorithms involving a wide range of applications. Gordon Hughes's "Henon Mapping with Pascal" demonstrates how to map complex mathematical functions that describe the behavior of dynamic systems such as asteroids, satellites, and charged particles.

The use of graphics does not always have to serve a specific application, however. We'll explore the domain of purely free-form recreational graphics. In "Abstract Mathematical Art," Kenneth E. Perry uses a BASIC program to produce cellular automata, the mathematical relatives of the Game of Life, and Peter B. Schroeder's "Plotting the Mandelbrot Set" examines the fractal geometry of nature in stunning displays on the Amiga personal computer.

The advent of high-performance, dedicated graphics microprocessor chips will provide the thrust for the next generation of graphics-oriented personal computers. Carrell R. Killebrew Jr. provides an inside look at the Texas Instruments TMS34010 Graphics System Processor. In the very near future the new generation of graphics engines will endow desktop machines with capabilities previously limited to large mainframe computers.

Engineers and scientists have a longstanding interest in plotting mathematical equations in three dimensions. "Graphing Quadric Surfaces" by George Haroney includes short BASIC routines that generate three-dimensional plots of equations in multiple colors. Steve Enns's "Free-Form Curves on Your Micro" explains how your personal computer can produce Bézier and B-spline curves that are normally confined to CAD/CAM and other high-end design applications.

All but one of the Theme articles include source code so you can experiment with these algorithms on your own personal computer and experience firsthand the fascination of computer graphics.

—*Charles Weston, Technical Editor*

PLOTTING THE MANDELBROT SET

BY PETER B. SCHROEDER

The more interesting the display, the longer it takes to generate

SOME PEOPLE CAN READ a musical score and in their minds hear the music more gloriously rendered than by any orchestra. Others can see, in their mind's eye, great beauty and structure in certain mathematical functions and number systems. Lesser folk, like me, need to hear music played and see numbers rendered to appreciate their structures.

The Mandelbrot set is one of the intriguing mathematical structures you can explore with the Commodore Amiga. In *The Fractal Geometry of Nature* (reference 1), Benoit Mandelbrot defines a fractal as "a set for which the Hausdorff Besicovitch dimension [fractional dimension] strictly exceeds the topological dimension."

You can use fractals to describe the fragmentary aspects of nature. Random fractals can simulate landscapes and objects in nature, such as trees and mountains. Mathematical modelers can use them to simulate shoreline decay and its effect on fisheries. Biochemists can use fractals to characterize the irregularity of protein surfaces and its influence on molecular interactions. You can use them to model climate and predict other apparently random and chaotic events (reference 2). You can use them to discover order where no order was previously believed to exist. Even processes of ontogeny—the course of an individual organism's development—that have proved elusive might be explorable using fractal geometry.

You need the definitions of complex, real, and imaginary numbers as a starting point. A complex number is any number of the form $a + bi$ where a and b are real numbers, and $i = \sqrt{-1}$. A real number is one in which there is no imaginary part, and an imaginary number is a complex number with b not equal to zero; some definitions require also that $a = 0$.

Engineers and others dealing with vector mathematics find imaginary numbers useful. For example, if vector b can be turned 180 degrees to point in the opposite direction by multiplying it by -1 so b equals vector $-b$, and there is a number, a, that turns the vector 90 degrees, then a^2 will turn the vector 180 degrees and be equal to -1. Thus, $a = \sqrt{-1}$.

WHAT IS THE MANDELBROT SET?

The Mandelbrot set is a set of numbers $z = c^2 + c$ where c is a complex number of the form $a + bi$ and z iteratively squared never produces a square root of $a^2 + b^2$ larger than 2. Note that since i^2 equals -1, $(a + bi)^2$ equals $a^2 + 2abi - b^2$, and that the iterative squaring of these numbers produces a jagged, nondifferentiable result. If the sum of the squares does grow beyond 4 within a large number of iterations, it will eventually approach infinity and, by definition, not be part of the Mandelbrot set.

If you take a matrix, a by b, and iteratively square every element in it until either the sum of their squares exceeds 4 or you reach 1000 iterations, you can determine a count of the number of iterations that each element in the array requires. Those elements with counts of 1000 are part of the Mandelbrot set; those elements with counts that are very large but still less than 1000 are near the Mandelbrot set; and those with low counts

(continued)

Peter B. Schroeder is president of Biosystems Research Inc. (11550 Southwest 108 Court, Miami, FL 33176). He is a biological consultant specializing in wetland and estuarine environments. He has a Ph.D. in biology from the University of Miami in Coral Gables.

are far from it. If you assign colors to each count or range of counts and color each pixel of the matrix display accordingly, you can graphically display the Mandelbrot set and its neighboring numbers. In a sense, the varying colors represent the fractal dimension distance from the Mandelbrot set for particular matrix elements.

SMALLER IS BIGGER

Fractal numbers have funny qualities. If you magnify areas near the Mandelbrot set, you will find minute imperfect replicas of the larger Mandelbrot set. If you increase the magnification even more, you find even more minute associated replicas, and so on ad infinitum. These components have great beauty and exotic structures, every one varying but always partially replicated at some smaller level.

Photo 1: *One instance of the Mandelbrot set representing the numbers from* −2.0 *to* 0.5 *on the x axis and from* −1.25 *to* 1.25 *on the y axis.*

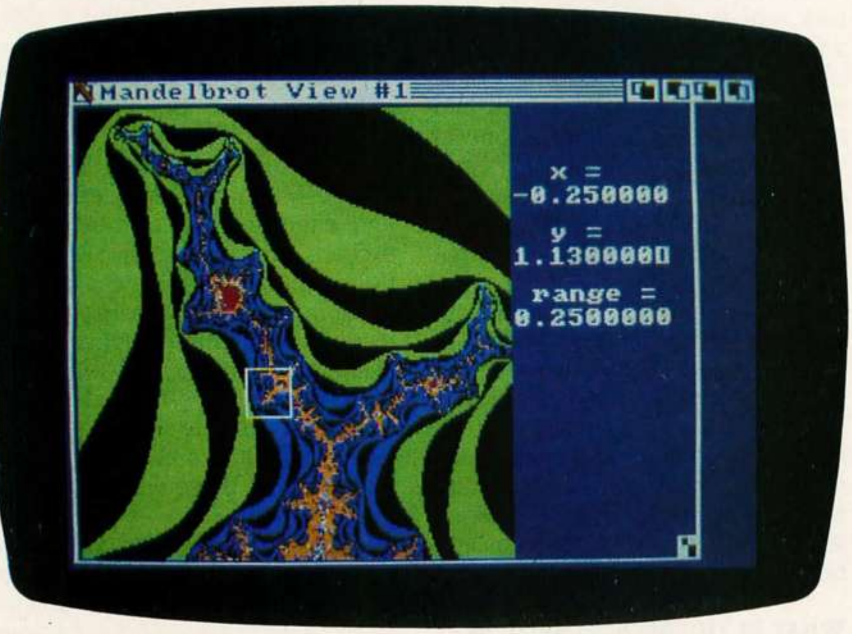

Photo 2: *An enlargement of the detail in the small box in photo* 1.

The magnification you can attain is limited only by your computer's precision. In C, the Amiga stores double-precision floating-point numbers in 64 bits with a precision, therefore, of approximately 2^{63}. It does all its math in double-precision floating-point, and its range for long integers is from −2,147,483,648 to 2,147,483,647 and for double-precision floating-point numbers from $\pm 10.0^{-307}$ to $\pm 10.0^{308}$ with 15 to 16 decimal digits of precision. This precision lets you examine the Mandelbrot set in greater detail than an electron microscope can examine matter. An electron microscope can produce a photo of an object enlarged about 1,000,000 times. You need a magnification between 300,000 to 400,000 times to resolve the facets on the surface of the poliomyelitis virus. If the Amiga were a microscope, its magnification would let you examine the atoms making up those facets.

Areas of interest are those closely associated with the larger Mandelbrot set in photo 1, which represents the numbers from −2.0 to 0.5 on the *x* axis and from −1.25 to 1.25 on the *y* axis. Its range is therefore 2.5 along both axes. Photos 2 and 3 are successive enlargements of the small outlined area in photo 1, each with a range 1/20 that of the preceding photo. The rectangular outlined area in all the photos shows the area enlarged in the succeeding photo.

THE PROGRAMS

A. K. Dewdney provided the algorithm for describing the Mandelbrot set graphically in *Scientific American* (see reference 3). My programs implementing that algorithm are written in Lattice C; they produced photos 1 through 3. While they take a while to run, they seem brisk compared to a BASIC implementation I tried first. [*Editor's note:* DOSET.C *and* VIEWSET.C *are available in* Lattice C *source code on disk, in print, and on* BIX; *see the insert card*

Listing 1: *The program DOSET.C, which calculates the Mandelbrot set.*

```
#include <stdio.h>
#include <math.h>

main(argc,argv) int argc;char *argv[];
{
 int y,x,count,totct;
 float x_coord,y_coord,range,gap,size,a,b,ac,bc,b1;
 char ct[201][2];
 FILE *OutFile;
  /* Input x-y coordinates and range from keyboard      */
  printf("Input X_COORDINATE: ");
  scanf("%f",&x_coord);
  printf("Input Y_COORDINATE: ");
  scanf("%f",&y_coord);
  printf("Input        RANGE: ");
  scanf("%f",&range);
  gap = range / 200.0;      /* Increment per pixel      */
  y_coord += range;         /* Start at top of display */
  /* Open output file       (default or command line)  */
  if(argc<=1) OutFile = fopen("df1:ZOOM.DATA","w");
  else OutFile = fopen(argv[1],"w");
  /* Write coordinates and range to data file           */
  fprintf(OutFile,"%7.6f\n",x_coord);
  fprintf(OutFile,"%7.6f\n",y_coord);
  fprintf(OutFile,"%8.7f\n",range);
  /* Calculate count value for each pixel (200X200)    */
  for(y=1;y<=200;y++)             /* Each row            */
  {
   bc = y_coord - y*gap; totct = 0;
   for(x=1;x<=200;x++)         /* Each pixel per row  */
   {
    ac = x*gap + x_coord;
    a = ac; b = bc; size = 0.0; count = 0;
    while(size < 4.0 && count < 1000)
    {
     /* Do complex-number multiply                      */
     b1 = 2*a*b;
     a = a*a - b*b + ac;
     b = b1 + bc;
     /* Pythagorean theorem                             */
     size = a*a + b*b;
     /* Don't need square root                          */
     count++;
    }
    totct += count;
    /* Code count in two bytes to save disk space       */
    ct[x][0] = count/256;
    ct[x][1] = count % 256;
   } /* End x loop                                       */
   /* Show row number and average count to CRT          */
   printf("%5d  %5d\n",y,totct/200);
   /* Print coded pixel-values this row to data file */
   for(x=1;x<=200;x++)
   {
    putc(ct[x][0],OutFile);
    putc(ct[x][1],OutFile);
   }
  } /* End y loop                                        */
  fclose(OutFile);  /* Close data file                   */
 } /* End main                                           */
```

following page 320 for details. You will need Amiga's translator.library, intuition.library, graphics.library, and narrator.device, as well as the Lattice C compiler, to run these programs. Listings are also available on BYTE*net; see page 4.*]

DOSET.C (listing 1) lets you pick the x,y axis—where you want to begin the display—and supply the distance you

(continued)

Photo 3: *An enlargement of the detail in the small box in photo 2.*

want to cover. The smaller this distance, the greater the magnification. These programs can take a long time—hours, even days—to generate a display. The closer you are to the Mandelbrot set, the longer it takes.

You should pick an *x,y* axis near the edge of the Mandelbrot set (colored red in the photos) for further exploration. If you pick a spot totally within the set, the display will seem to take forever—not literally, since the programs stop counting after 1000 iterations at each pixel—and the display will be a solid color. If you pick a spot where each pixel takes the full 1000 iterations (i.e., within the set proper) and you generate a 200- by 200-pixel display, the Amiga has to carry out 40 million iterations. If an iteration does approximately 20 double-precision floating-point operations, that's 800 million double-precision (64-bit) operations. And if each operation requires manipulating all the bits, that's 51 billion bit operations. On the other hand, if you start far away from the Mandelbrot set, you will finish a lot sooner, but you end up with a correspondingly uninteresting display. So choose a display that includes an area alongside, and maybe including a bit of, the set proper.

VIEWSET.C creates a 200- by 200-pixel graphics display from the data file created by DOSET.C. If you want to pursue this further, try changing the color separation in the program. With the proper color separation, you can find all sorts of interesting objects: trees, jabberwocks, maybe even a self-portrait—or millions of them. Since the Amiga can show up to 640 by 400 pixels, you can also create high-resolution displays. However, these displays take even longer to generate.

New Horizons

If you do look into the Mandelbrot set, happy exploring. The going may be slow, but the region is huge and will always remain largely unexamined. Because of the time involved, if you just want to look at the displays I have already generated, I will mail you a disk full of them at my cost ($10). You may share it as you wish. ■

References

1. Mandelbrot, Benoit B. *The Fractal Geometry of Nature.* New York: W. H. Freeman and Company, 1983.
2. Nicolis, C., and G. Nicolis. "Is There a Climatic Attractor?" *Nature,* October 1984, pages 529–532.
3. Dewdney, A. K. "Computer Recreations: A Computer Microscope Zooms in For a Look at the Most Complex Object in Mathematics." *Scientific American,* August 1985, pages 16–24.