Thomas A Dwyer
Margot Critchfield
Project Solo
University of Pittsburgh
Pittsburgh PA 15260

Color Graphics on the Compucolor 8051

The ancient wisdom that says "a picture is worth a thousand words" has a special significance for the computer age. With machines that can generate output faster than anyone can read it, there's no doubt that we need new ways to represent this avalanche of data. The best answer (so far) seems to be in computer graphics: sophisticated pictures that show the results of all this computation in a form that is easy to interpret and even easier to remember.

One of the most dramatic ways to improve graphical output is to add color. Color graphics conveys information to human viewers that is hard to appreciate until it's experienced. The change from black and white to color is at least as impressive as the change from monaural to full stereo sound.

Until a few years ago, the hardware for color graphics was both rare and expensive. This situation is changing, however, mainly through the efforts of manufacturers who have devised ingenious ways to use the technology of color TV in conjunction with computer technology.

One of the first products designed for this purpose was the Cromemco TV Dazzler. It consists of two Altair (S-100) compatible boards with a coaxial cable output to a color monitor. Several examples of color art produced with the TV Dazzler were shown in December 1976 BYTE. That issue also contained the article, "The Cybernetic Crayon," that gave an example of an 8080 assembly language program for the Dazzler.

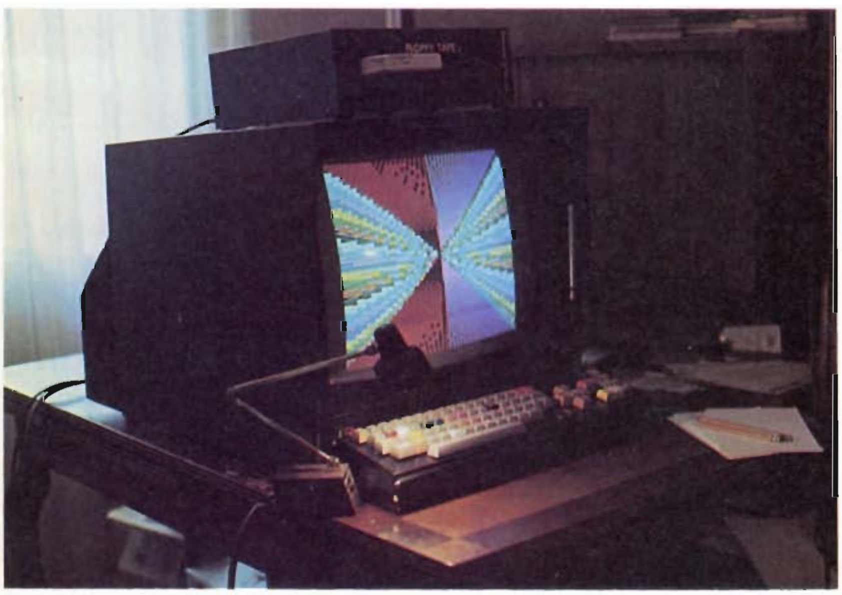


Photo 1: The Compucolor 8051 with extended ASCII keyboard and "floppy tape" storage. The tapes are similar to 8 track music cartridge tapes, except that they are shorter. More recent models use a floppy disk system for storage.

The Compucolor 8051

A more recent example of personal color systems is the Compucolor 8051. It comes in the form of a complete computer system packaged in a cabinet not much bigger than a 19 inch color television. That's impressive because the cabinet contains the processor, memory, interface, programmable read only memory, graphics controller, power supplies, and a 19 inch color display tube. The colors are of much greater purity than those usually seen on TV because the system uses a very wide bandwidth (75 MHz) and the three color circuits (red, green, blue) are kept separate. This is the same approach used in professional color monitors. Since any of these colors can be on or off, eight colors (including black) are available.

Photo 1 shows a typical system consisting of the computer and color display in the large cabinet, a keyboard for input, and a dual 8 track tape cartridge unit for mass storage.

The large screen holds 48 lines of 80 characters each, about four times the number of lines found on most video displays. When characters are used for graphics, there are 3840 plotting positions available. But a finer division of each character into a 4 by 2 array of "points" is also possible, as figure 1 indicates. In point plot mode, one character block becomes eight "point" positions. Each of these points (which is really a small rectangle) can be turned on or off separately. This means there are 160 point positions for the horizontal direction (X axis), and 192 points for the vertical direction (Y axis). So a total of 30,720 points can be plotted.

A special hardware feature called "vector graphics" is also available. This makes it possible for the computer to draw approximations to straight lines on the screen if you simply tell it where to start (X0, Y0), and where to finish (X1, Y1). The hardware calculates all the point positions in between and plots them automatically.

The Compucolor 8051 comes with a BASIC interpreter stored in read only memory. This means that you never have to load BASIC: you call for it by just pressing two keys (Escape, followed by W).

The BASIC interpreter has all the standard features and a few extra ones (including string arrays). It also has the special key word PLOT. This feature makes it easy to program graphics of all kinds in color. Let's look at some examples of how it works.

Using Compucolor BASIC

The word PLOT is used in BASIC statements of the form

```
25 PLOT I
```

where I is an integer from 0 to 255. When PLOT I is initially used, the value of I determines what *mode* you enter. There are quite a few modes. The main mode is "point plot" (PLOT 2). After you enter a "mode," the meaning of PLOT I is different, and it depends on what mode you're in.

Example 1: Suppose we want to plot the curve $Y=X^2$. We first use PLOT 2 to put the program in point plot mode. After saying PLOT 2, each pair of PLOT statements that follow will then give the X and Y positions of the point to be plotted. When finished plotting points, we use PLOT 255 to escape from the plot mode we're in. Here's a program that plots 100 points of our curve:

```
10 PLOT 2
20 FOR X = 0 TO 99
30 LET Y = INT(X*X/100)
40 PLOT X:PLOT Y
50 NEXT X
60 PLOT 255
```

To understand the output of this program you have to know that the "origin" for point plot mode is the point (0, 0), and that it is located in the lower left corner of the screen. All points (X, Y) must be described

Figure 1: Division of screen character into a 2 by 4 array of "points."

by positive integers with X going from 0 to 159, and Y going from 0 to 191. To make Y=X*X fit on the screen, we divided by 100 (because when X=99, Y=9801 which would be way off the screen). Our graph will look something like the one in figure 2.

Example 2: Here is a program to plot a sine function. We let X go from 0 to 159, but actually plot SIN(X/13). This causes the argument of SIN to go from 0 to slightly over 12 radians, giving about two cycles of the function. Since the SIN function has values from −1 to 1, we multiply it by 95 and then add 95. This causes Y to go from 0 to 190. The program is initiated by the command PLOT 12, which means "clear screen."

```
10 PLOT 12:PLOT 2          'Clear screen, enter plot mode
20 FOR X=0 TO 159          'Get X
30 Y=95*SIN(X/13)+95       'Calculate Y
40 PLOT X: PLOT Y          'Plot X, Y
50 NEXT X                  'Back to line 20
60 PLOT 255                'Exit plot mode
```

Adding Color

The preceding programs plot in whatever color the machine was using last. Color can be changed with either the PLOT 6 or the PLOT 29 modes. After entering the PLOT 6 mode, the statement PLOT I produces various combinations of foreground colors, background colors, and "blink" states, all depending on the value of I. For example, I = 0 to 7 gives the eight possible foreground colors on a background of black with no blink.

The PLOT 29 mode is a little simpler. It allows only foreground (which means the actual point being plotted) color to be specified. It is followed by PLOT I, where I specifies color according to the codes: 16=black, 17=red, 18=green, 19=yellow,

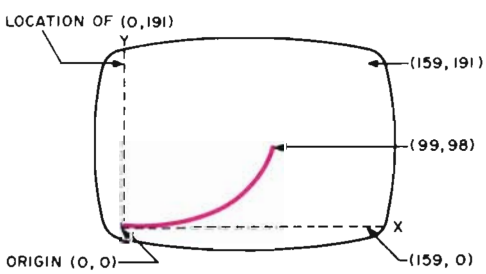


Figure 2: A display of the equation $Y=X^2$.

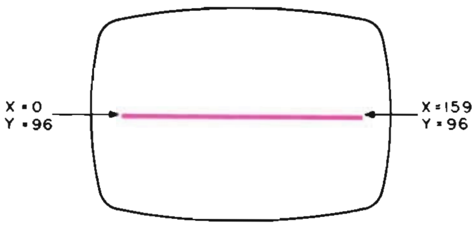


Figure 3: A display of the line running from points (0,96) to (159,96).

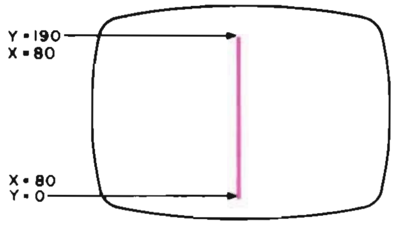


Figure 4: A display of the line running from points (80,0) to 80,190).

```
10   PLOT 12                                'Erase screen
20   PLOT 29: PLOT 19                       'Set color= yellow
30   PLOT 2                                 'Go into main plot mode
40   PLOT 246: PLOT 0: PLOT 0: PLOT 190     'Plot Y bar graph
50   PLOT 250: PLOT 0: PLOT 95: PLOT 159    'Plot X bar graph
60   PLOT 255                               'Exit plot mode
70   FOR X = 0 to 159                       'Get X
80   PLOT 29: PLOT 17+X/22                  'Set a color depending on X
90   Y = 95*SIN(X/13)+95                    'Calculate Y
100  PLOT 2                                 'Point plot mode
110  PLOT X: PLOT Y                         'Plot one point
120  PLOT 255                               'Exit plot mode
130  NEXT X                                 'Back to line 70
```

Listing 1: A BASIC program for graphing a sine function in seven different colors.

20=blue, 21=violet, 22=cyan (light blue), and 23=white.

We'll illustrate this feature in a moment, but let us first look at an example of a subplot mode (a mode that follows the PLOT 2 main mode).

Bar Graph Mode

After entering PLOT 2 mode, you can go into several subplot modes. For example, PLOT 250 means enter the X (horizontal) bar graph mode. It is followed by three PLOT statements that tell where to put the bar graph and how long to make it.

```
PLOT 2: PLOT 250
PLOT 0: PLOT 96: PLOT 159
```

This directs the program to draw a horizontal bar (line) from X=0 to X=159, 96 units up on the screen (the Y position). See figure 3. The code for Y (vertical) bar graphs is PLOT 246.

```
PLOT 2: PLOT 246
PLOT 0: PLOT 80: PLOT 190
```

The latter draws a vertical bar (line) from Y=0 to Y=190, 80 units to the right (the X position). See figure 4.

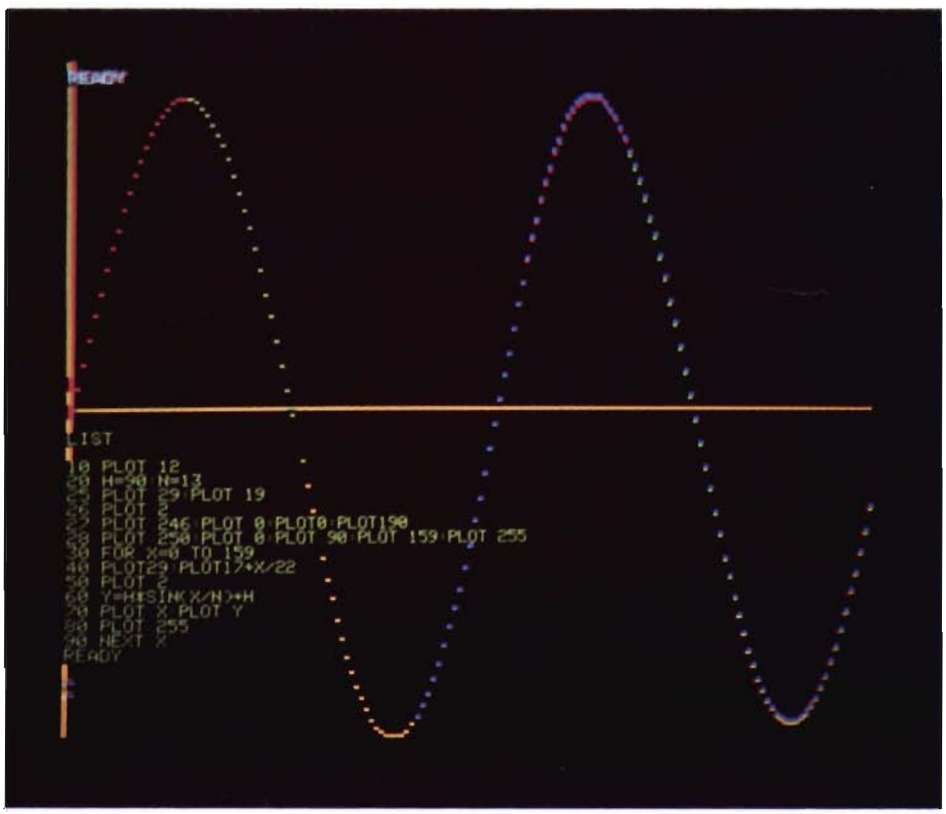


Photo 2: A plot of the sine function, which changes the color of the plotting point cyclically. The axes are drawn with bar graph mode.

A Combined Program

The program shown in listing 1 graphs the sine function using seven different colors (codes 17 to 23). It also uses the bar graph mode to plot the X and Y axes for the graph. The axes are drawn in yellow (code 19). The output of this program is shown in photo 2. Notice in line 100 that PLOT 2 serves a dual purpose. It puts the machine in both main and point plot mode. A more interesting example of the output possible from bar graph mode is shown in photo 3.

Vector Mode

The bar graph modes are used to draw horizontal and vertical lines. Vector mode allows you to draw lines (called vectors) between any two points. This means slanted lines can be drawn on the screen. Start with PLOT 253, then give the coordinates of the starting position with PLOT X0:PLOT Y0. The next command should be PLOT 242, followed by the coordinates of the end position with PLOT X1:PLOT Y1. (Any variable names can be used; we find these easy to remember.) The program in listing 2 lets you draw vectors on the screen using seven colors in sequence. (Black is not used because it would draw an "invisible" line.)

If you wish to draw lines where the end point of one is the starting point of the next one (as in drawing polygons), you need only use PLOT 253 once. The program in listing 3 draws random vectors in this fashion, making an attractive abstract type of drawing. The output is shown in photo 4.

Incremental Plot

Another submode that can follow PLOT 2 is the PLOT 251 incremental plot, which allows you to move the graphic "point" element (really a small rectangle) by a small "increment" (or step). There are eight directions in which you can move, shown by the arrows in the following diagram:

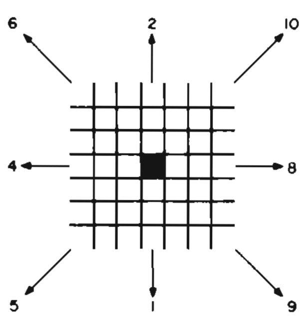


Photo 3. A display of concentric rectangles produced in the bar graph mode.

```
10  PLOT 12:C=0                        'Erase screen
20  C=C+1                              'Start with color code 1
30  IF C > 7 THEN C=1                  'Only allow seven color codes
40  PLOT 6:PLOT C                      'Set the color
50  PRINT"TYPE X0, Y0";
60  INPUT X0, Y0                       'Get starting point
70  PRINT"TYPE X1, Y1";
80  INPUT X1, Y1                       'Get end point
90  PLOT 2                             'Go into plot mode
100 PLOT 253:PLOT X0:PLOT Y0           'Draw vector starting at X0, Y0. . .
110 PLOT 242:PLOT X1:PLOT Y1           '. . .to X1, Y1
120 PLOT 255                           'Exit plot mode
130 GOTO 20                            'Repeat
```

Listing 2: A BASIC program enabling the user to draw vectors on a color video screen using seven colors in sequence.

```
10 K=16                                'Initialize color selector
20 PLOT 12:PLOT 2                      'Enter plot mode
30 PLOT 253:PLOT 79:PLOT 91            'Start vector in center
40 K=K+1                               'Select color . . .
50 IF K>23 THEN K=17                   '. . .between 17 and 23
60 PLOT 255:PLOT 29:PLOT K             'Turn on color
70 X=160*RND(1) :Y=190*RND(1)          'Choose random point
80 PLOT 2:PLOT 242:PLOT X:PLOT Y       'Draw vector
90 GOTO 40
```

Listing 3: A BASIC program that draws a chain of random vectors.

```
10  PLOT 12                            'Erase screen
15  PLOT 2:PLOT 80:PLOT 92             'Plot a point in middle of screen
20  FOR C=17 TO 23                     'Select a color
30  PLOT 255:PLOT 29:PLOT C            'Activate color
40  PLOT 2.PLOT 251                    'Go into incremental plot mode
50  E = 2 ↑ INT(4*RND(1) )             'Choose direction
60  FOR K=1 TO 5*RND(1)+5              'Choose number of steps
70  PLOT E                             'Plot in direction E
80  NEXT K                             '    for K steps
90  NEXT C                             'Get next color and direction
100 GOTO 20                            'Repeat the whole cycle
```

Listing 4: A BASIC program to produce a random walk pattern.

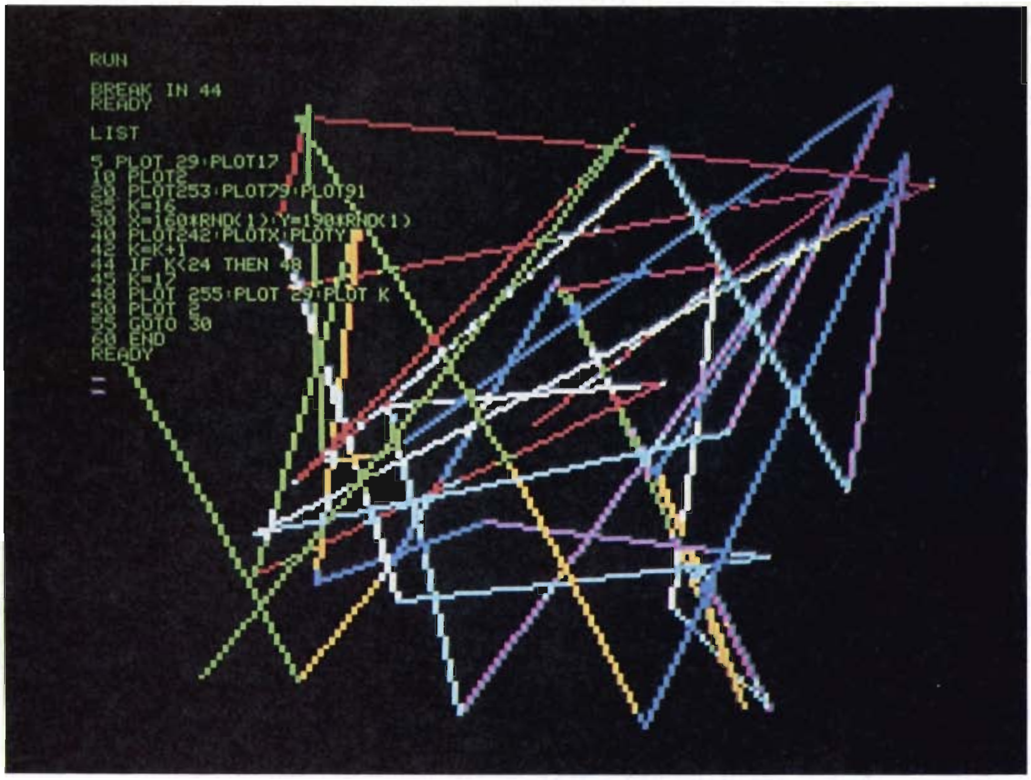


Photo 4: A random vectors program display. Color is changed cyclically for each new vector. The listing overwrites some of the vectors because it was done after interrupting the run.

The numbers next to the arrows are "direction codes." A given direction is selected by using the proper direction code in a second PLOT statement. For example, to plot a point in the middle of the screen and then move it one "increment" to the right, use the following instructions:

```
10 PLOT 2:PLOT 80:PLOT 92
20 PLOT 251:PLOT 8
```

PLOT 251 means "increment (move) the point," and PLOT 8 means "to the right." The distance moved is very small (the width of one point).

The next program (see listing 4) uses this feature to produce what is called a random walk pattern. We'll use only the direction codes 1, 2, 4 and 8. These will be generated by the formula

```
E = 2 ↑ INT(4*RND(1))
```

So we randomly get the numbers $2^0=1$, $2^1=2$, $2^2=4$ and $2^3=8$. Our point will randomly "decide" to move down, up, left and right.

To make the pattern even more interesting, we use another random number (from 5 to 9) to decide how many increments (small steps) to take in each direction. We also change colors. The result is a rather striking weave-type color abstraction. (Actually, random walk techniques originated for practical reasons. They are used in solving several types of mathematical and scientific problems that can't be handled by conventional methods.) Line 50 selects a random direction, and the loop 60-80 determines how many increments to take in that direction. The main loop 20-90 repeats the process for seven different colors. The program goes on indefinitely, filling the screen to any density you wish. Photo 5 shows the output of this program after

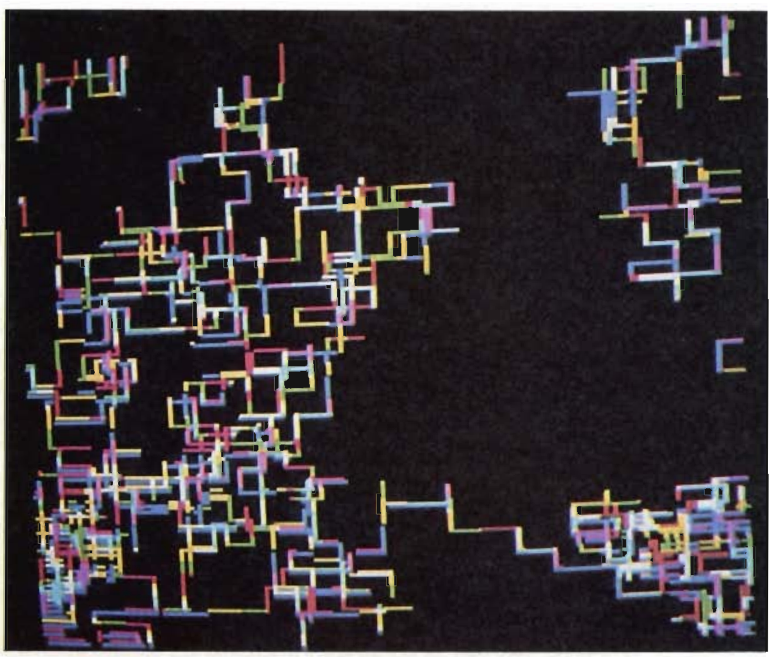


Photo 5: Output from the random walk program after 20 seconds. When the plotting point "walks off" the edge of the screen, it reappears on the opposite side due to wraparound in the display memory.

Photo 6: A random walk after 20 minutes. The plotting point moves rapidly, but it frequently overwrites itself. This program can be viewed as a striking way to study random number generators.

20 seconds. Photo 6 shows the output after about 20 minutes. The plotting is rapid, but paths are often retraced.

There are several other subplot modes available, including an incremental vector mode. This has the effect of shifting a vector to a new position very close to the starting position. Since vectors are drawn very rapidly (output to the video display is at 9600 bps), this mode has interesting possibilities for producing simple animation.

A light pen option is also available; it can be used as an input device in developing color graphics. For example, a program in BASIC can be written to accept (X, Y) coordinates from the pen and use these to determine the position of one end of a vector. If the other end of the vector is determined by an algorithm within the same program, the graphic "artist" can interact with the algorithm to produce variations on a design. Photo 7 shows a simple example of this. The outer set of points was determined by an algorithm which generates a 2 by 3 Lissajous figure, while the inner set of points was selected with the light pen. Photo 8 shows the effect of running the same program, having first keyed in a solid blue background. Vectors

Photo 7: A pattern produced with the aid of a light pen. It consists of varicolored vectors; the outer set of end points is determined by a Lissajous figure algorithm. The inner set of end points is selected with a light pen.

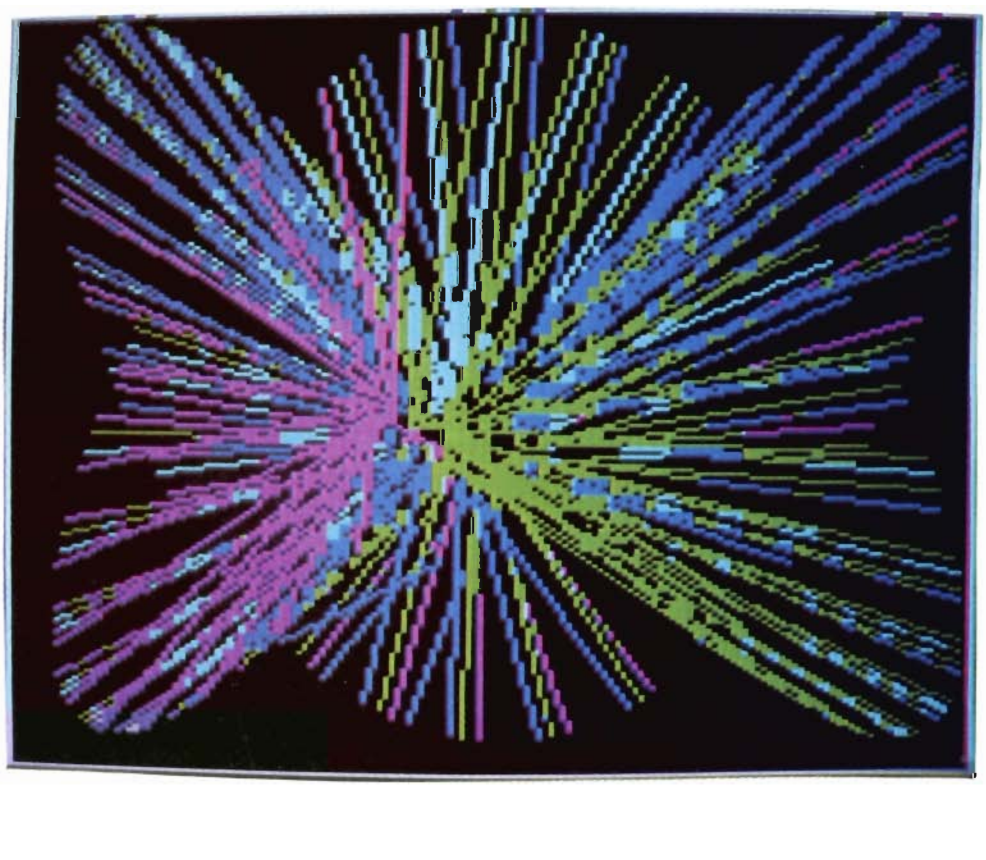
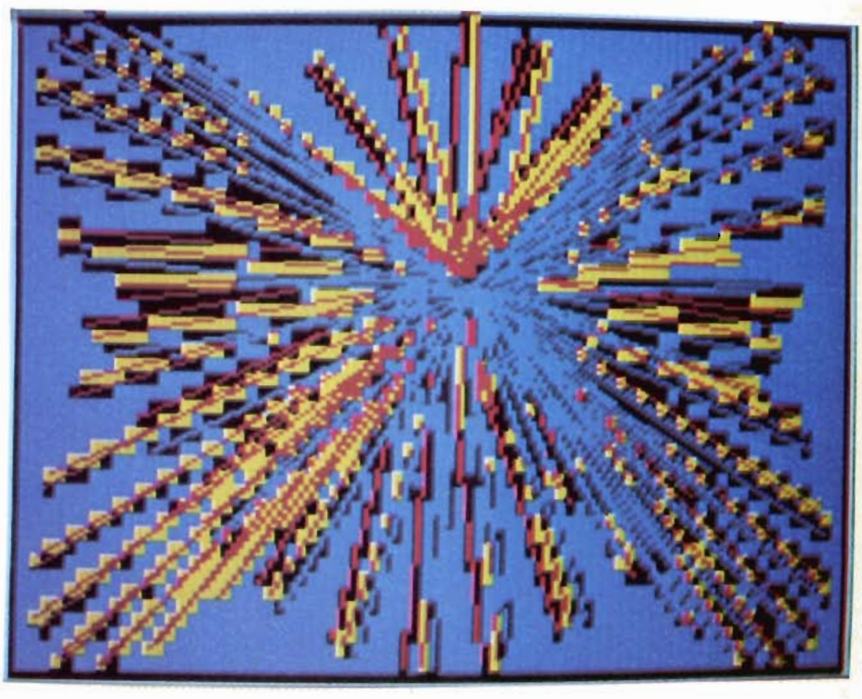


Photo 8: A pattern produced in the same manner as that of photo 7, except that the background color has first been set to blue. Background was reset to black to draw some of the vectors, and to blue for others. The artist used the light pen to draw a sprinkling of "blue on blue" vectors over the final pattern to produce a "burst" effect.



were drawn in "color-on-black" and in "color-on-blue" modes. A general program for experimenting with the light pen in this manner is given in the book *BASIC and the Personal Computer* (Addison-Wesley, Reading MA 01867).

Given the value of output in the form of dynamic as well as color graphics, and the complex ideas which can be programmed in Compucolor BASIC, the Compucolor 8051 is quite an interesting machine. Its design invites explorations well beyond those described in this brief introduction.■