BITMAPPED VERSUS STRUCTURED GRAPHICS

Video display memory contains bitmaps by nature. The actual graphics files may or may not be bitmaps. The other general file category is structured graphics. The differences are significant, and each has its own pluses and minuses.

◆ Bitmapped Graphics

Bitmapped graphics use one or more bit planes to represent the value of each pixel in an image file. Bitmapped graphics are often referred to as raster graphics, because the bitmap in the display buffer directly corresponds to the raster lines on the CRT. As such, they are the predominant type of graphic files used for multimedia presentations.

As discussed, bitmaps aren't necessarily the same resolution as the screen. The more resolution that is used to represent an image, the smoother the curves and other visual elements will appear when output to a device capable of rendering a matching resolution.

Fixed image resolution is fine when working with a device with the same resolution, but it has drawbacks when output or displayed on devices with varying resolution. Images whose actual resolution exceeds the screen resolution are inappropriate for final presentation on monitors. On the other hand, most film recorders benefit from higher resolution when making slides and transparencies.

While image processing programs can alter the resolution of an image, resolution-dependency is a major drawback in manipulating bitmaps. Their literal description of an image makes for imperfection in artistic tasks such as scaling and perspective. Such operations yield the greatest fidelity when performed with a conceptual knowledge of the image rather than a finite description.

Bitmapped images have fairly large file sizes. The greater the image resolution, image size, and color resolution, the more unwieldy the file becomes to store, manipulate, display, and animate. A 24-bit display with a size of 640 x 480 takes up about 921K before compression. The same size image at 300 dpi would require 3.8MB—and much larger image sizes and resolutions are common in the world of print production.

Some bitmapped file formats employ a form of automatic redundancy compression. Say you've drawn a horizontal black line 80 pixels long. Rather than storing 80 pieces of identical data, the software can store something like "color: black, occurrences: 80." In this manner, 80 pieces of data can be reduced to two or three. The Amiga IFF file formats are examples of built-in redundancy compression. (TIFF and TARGA offer redundancy compression as options.)

Jeff Burger, The Desktop Multimedia Bible (N.Y., NY: Addison-Wesley, 1993)

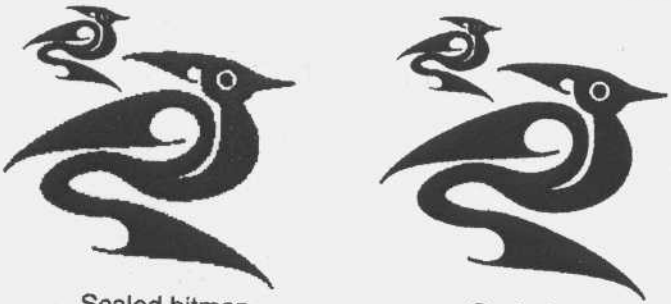
❖ Structured Graphics

The alternative to bitmaps is broadly termed *structured graphics*—also known as *object-oriented*, or *vector, graphics*. Rather than describing every dot in an image, these formats describe objects such as lines, circles, rectangles, and polygons. The descriptions typically include attributes like anchor points, size, angles, position, line weight, and hollow/fill characteristics. These commands are then rendered into a bitmap that matches the resolution of the output device (printer, monitor, film recorder). Structured graphics are said to be *device independent* or *resolution independent*—thereby obviating one of the problems of bitmapped graphics. (See Figure 7.17.)

Here's another way to look at structured graphics. A point represents a single conceptual position of infinitesimal size located on a plane. *Vector* refers to a conceptual line drawn between two points. You could draw that line as a series of individual dots with a pencil (high resolution), a felt-tip pen (medium resolution), or a magic marker (low resolution)—each yielding a physical representation with different resolution. The actual points and line are still conceptual: only their physical interpretation has changed.

Figure 7.17

The image descriptions provided by structured graphics provide much better scaling effects than do the literal descriptions provided by bitmaps.



Structured graphics overcome the problem of bitmapped file sizes. If you draw a big solid green circle that covers most of the screen, the computer's internal representation of that might be something like "draw circle: center 4" in, 3" down; radius 2.5"; fill green." That set of instructions takes a handful of bytes; a bitmap of the same image could range from kilobytes to megabytes depending upon the physical and color resolution.