

Computer Graphics

Dr./ Ahmed Mohamed Rabie

Chapter 2

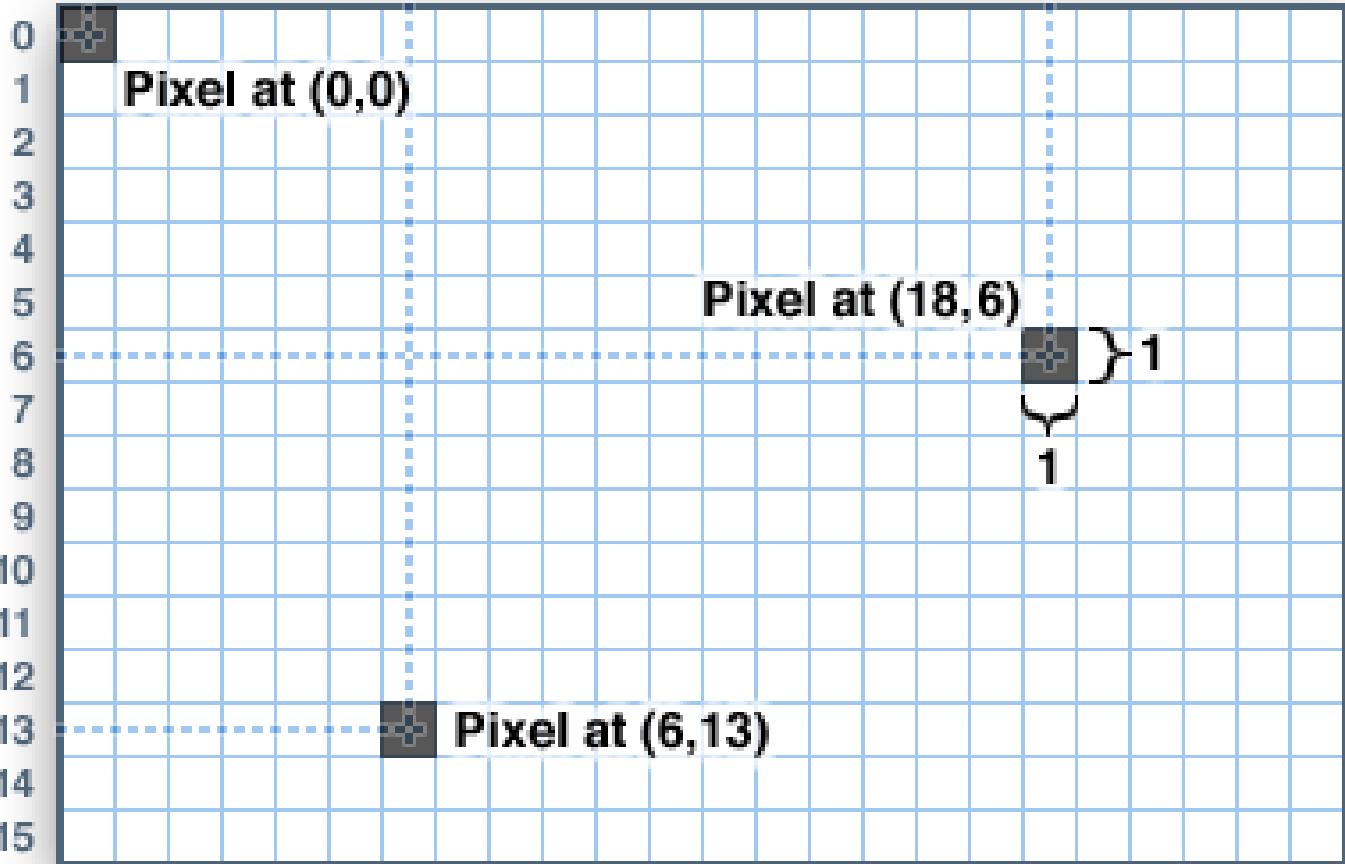
Two Dimensional Graphics

To create a **two-dimensional image**, each point in the image is assigned a color. A point in 2D can be identified by **a pair of numerical coordinates**. Colors can also be specified numerically. However, the assignment of numbers to points or colors is somewhat arbitrary.

A **digital image** is made up of rows and columns of **pixels**. A pixel in such an image can be specified by saying which column and which row contains it. In terms of coordinates, a **pixel** can be identified by a **pair of integers** giving the column number and the row number.

$+x$ 

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23



$+y$



Row and column numbers identify a pixel, not a point. **A pixel contains many points;** mathematically, it contains an infinite number of points. **The goal of computer graphics** is not really to color pixels—it is to **create and manipulate images.**

Bit depth is the number of bits used to store information about each pixel. The higher the depth, the more colors are stored in an image. For example, the lowest bit-depth, **1 bit graphics** are only capable of showing **two colors**, black & white. This is because there are only two combinations of numbers in one bit, 0 and 1. **Four bit color** is capable of displaying **16 colors** because there are 16 different combinations of four bits.

With **8 bit color**, there is a total of **256 colors available**.

With 16 bit color, a total of 65536 is available. When

you have 24 bit color, a total of 16777216 colors is

available. Bit depth might refer to the bit depth of an

image, or the bit depth of a monitor. When it refers to

the bit depth of a monitor, it means that your monitor

is able to show a certain number of colors at one time.

If it is meant as the bit depth of an image, it means that

^

the image may contain the specified number of colors.



1-Bit image (781 bytes) 8-Bit image (5,362 bytes) 16-Bit image (11,982 bytes)

The same image of different bit-depths. You might not be able to appreciate all the quality of 16 bit images if your monitors do not support those bit-depths. You might wonder why not use the best bit depth for all images to make them look good. Well, as you might have noticed, the higher the bit depth, the more bits per pixel used in an image, and the more pixels used in an image, the larger the actual file is going to be. Sometimes you don't need very high image quality, so you won't use a high bit depth to preserve space.

A **bitmap** is a collection of pixels that describes an image, in human terms, a complete picture. A bitmap can be of various bit depth and resolution. Basically, a bitmap is an array of pixels. A bitmapped graphic Enlarged to show detail and individual pixels.



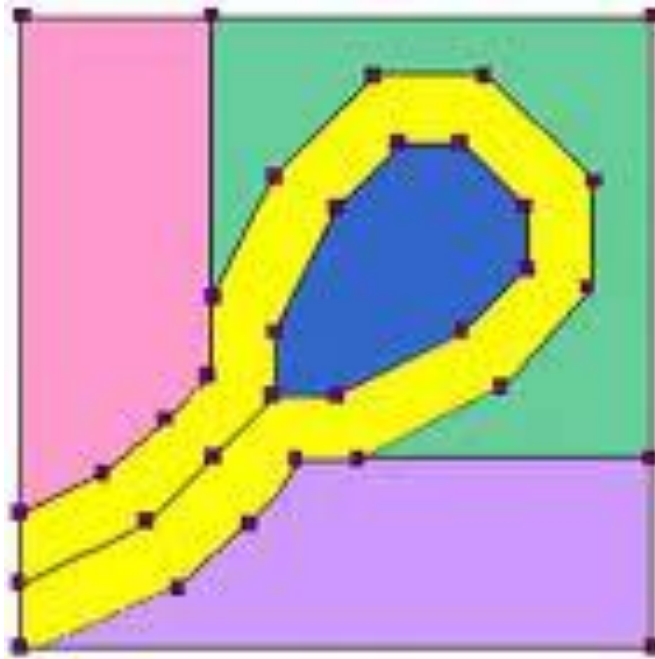
Resolution in computer graphics either refers to the number of pixels per inch or other unit of measure (centimeter for example) on a monitor or printer. It is also sometimes used to describe the total number of pixels on a monitor. Resolution is usually measured in pixels per inch or dots per inch (dpi).

Most monitors can display at 72 dpi. **Monitors come in a variety of resolutions.** They can get as small as 320 pixels by 200 pixels or as large as 1280 by 1024. Printers can also print at many different resolutions. The resolution can be as small as 128 dpi or less, to 300 dpi on inkjet printers, to 720 dpi on laser printer, and up to 2000 dpi on high quality type-setters.

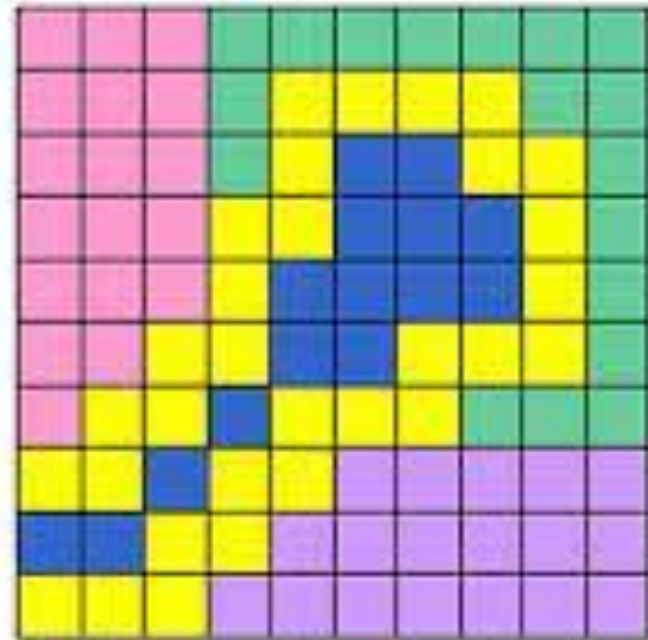
Both vector and raster graphics have their unique features such as the properties of graphic files and the way they are used. One of the main differences is the way in which a particular object is described. While **raster graphics uses pixels.**

Vector graphics is based on curves and mathematical shapes (circles, ellipses, polygons, etc.) defined by **mathematical equations**. Both raster and vector graphics can also be used at another graphic level.

Vector graphics is used for creating illustrations, diagrams, promotional materials (leaflets, business cards, and logotypes), etc.



Vector



Raster

Graphics displays often include **textural information**, such as labels on graphs and charts, signs on buildings or vehicles, and general identifying information for simulation and visualization applications. **Routines for generating character primitives are available in most graphics packages.**

Some systems provide an extensive set of character functions, while other systems offer only minimal

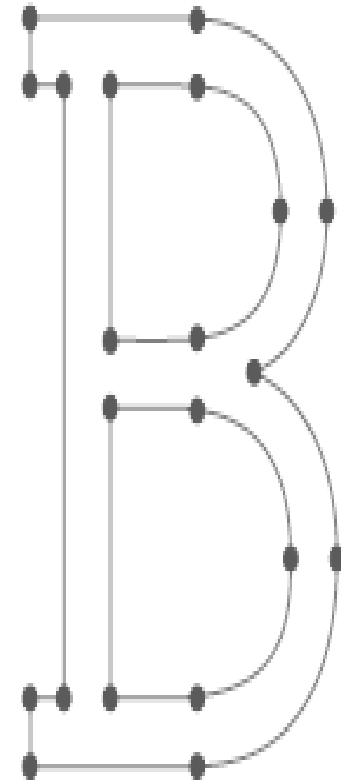
Two different representations are used for storing computer fonts. A simple method for representing the character shapes in a particular typeface is to set up a pattern of binary values on a rectangular grid. The set of characters is then referred to as a bitmap font (or bitmapped font).

A bitmapped character set is also sometimes referred to as a **raster font**. Another, more flexible, scheme is to **describe character shapes using straight-line and curve sections**, as in PostScript, for example. In this case, the set of characters is called **an outline font or a stroke font**.

The letter "B" represented with an 8×8 bitmap pattern (a) and with an outline shape defined with straight-line and curve segments (b).

1	1	1	1	1	1	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	1	1	1	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

(a)



(b)

Figure illustrates the two methods for character representation. When the pattern in Figure is applied to an area of the frame buffer, **the 1 bits designate which pixel positions are to be displayed in a specified color.** To display the character shape in Figure, the interior of the character outline is treated as a fill area.

Bitmap fonts are the simplest to define and display: We just need to map the character grids to a framebuffer position. In general, however, **bitmap fonts require more storage space because each variation (size and format) must be saved in a font cache.** It is possible to generate different sizes and other variations, such as bold and italic, from one bitmap font set, but this often does not produce good results.

We can increase or decrease the size of a character bitmap only in integer multiples of the pixel size. To double the size of a character, we need to double the number of pixels in the bitmap. This just increases the ragged appearance of its edges. In contrast to **bitmap fonts**, **outline fonts can be increased in size without distorting the character shapes**. And outline fonts require less storage because each variation does not require a distinct font cache.

We can produce boldface, italic, or different sizes by manipulating the curve definitions for the character outlines. But it does **take more time to process the outline fonts because they must be scan-converted into the frame buffer**. Some graphics packages provide a function that accepts any character string and a frame-buffer starting position for the string. Another type of function displays a single character at one or

۳۶ more selected positions.

Geometric descriptions for characters are given in world coordinates, just as they are for other primitives, and this information is mapped to screen coordinates by the viewing transformations. **A bitmap character is described with a rectangular grid of binary values and a grid reference position. This reference position is then mapped to a specified location in the frame buffer.**

An outline character is defined by a set of coordinate positions that are to be connected with a series of curves and straight-line segments and a reference position that is to be mapped to a given frame-buffer location. The reference position can be specified either for a single outline character or for a string of characters. In general, character routines can allow the construction of both two-dimensional and three-dimensional character displays.

Some graphics libraries include routines for describing a picture as a collection of named sections and for manipulating the individual sections of a picture. Using these functions, we can create, edit, delete, or move a part of a picture independently of the other picture components. Various names are used for the subsections of a picture. Some graphics packages refer to them as structures, while other packages call them segments or objects.

Also, the allowable subsection operations vary greatly from one package to another. **Modeling packages**, for example, provide a wide range of operations that can be used to describe and manipulate picture elements. On the other hand, for any graphics library, we can always structure and manage the components of a picture using procedural elements available in a high-level language such as C++.