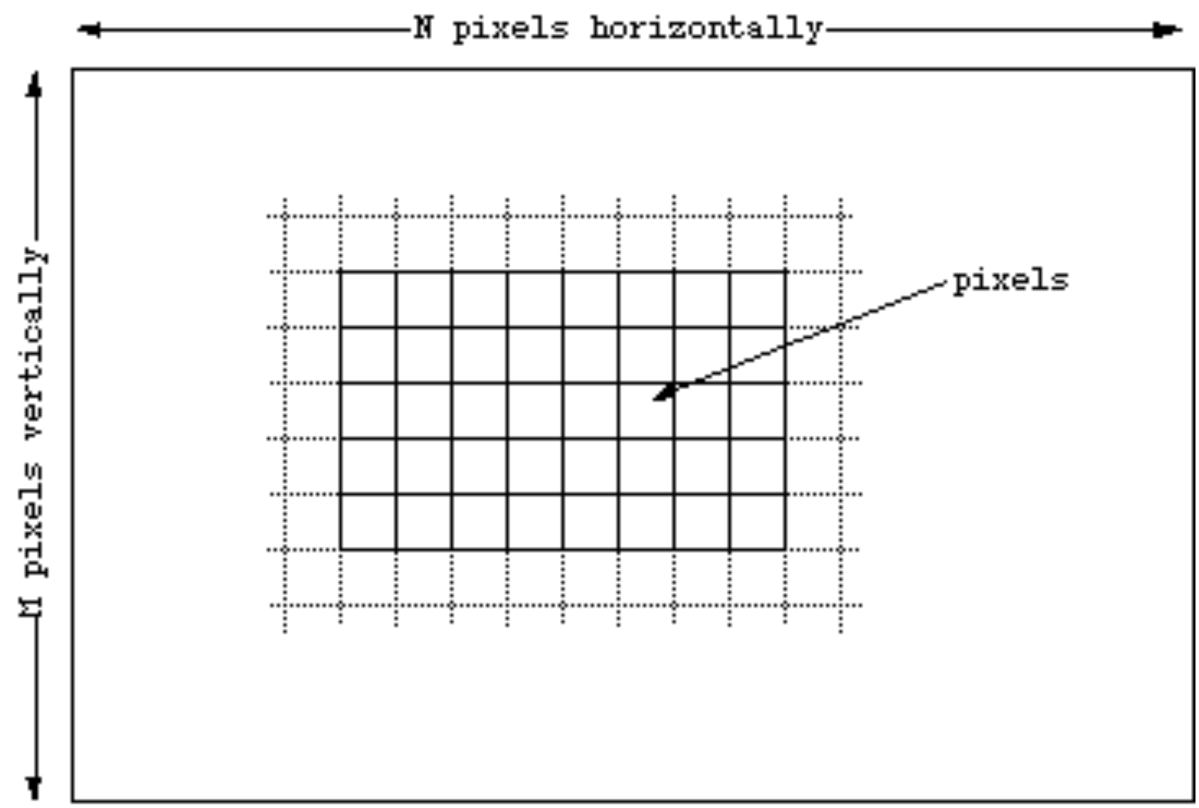
A Beginners Guide to Bitmaps

Bitmaps are defined as a regular rectangular grid of cells called pixels, each containing a color value. Bitmaps are characterized by two parameters, the number of pixels within the rectangular grid and the information content (color depth) per pixel. There are other attributes that are applied to bitmaps, but they are derivations of these two fundamental parameters.



The following image is a bitmap that has 397 pixels horizontally, 294 pixels vertically, and each pixel contains a grey value from a possible 256 different greys.

Color "depth"

Each pixel in a bitmap contains certain information, usually interpreted as color information. The information content is always the same for all the pixels in a particular bitmap. The amount of color information depends on the application requirements and are described below.

1-bit (black and white)

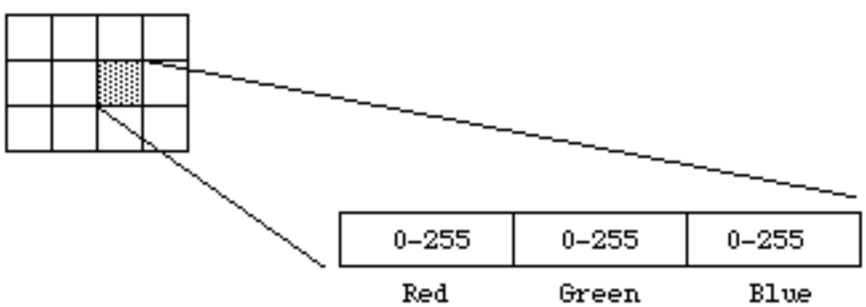
This is the smallest possible information content that can be held for each pixel. The resulting bitmap is referred to as monochrome or black and white. The pixels with a 0 are referred to as black, pixels with a 1 are referred to as white. Note that while only two states are possible, they could be interpreted as any two colors, 0 is mapped to one color, 1 is mapped to another color.

8-bit greys

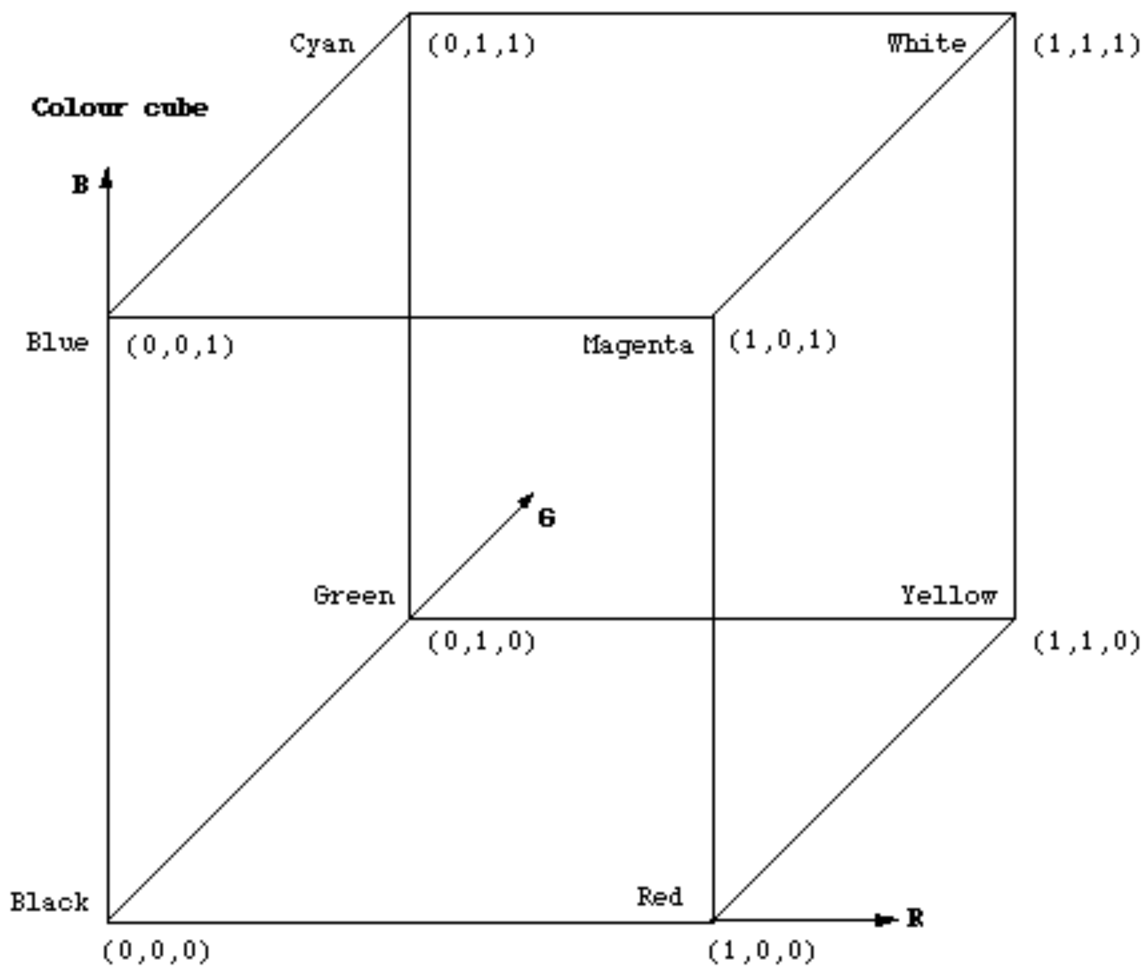
In this case each pixel is stored as 1 byte (8 bits) of information. Since 2^8 results in 256, this implies there are 256 different shades of gray, from black to white. This type of bitmap is referred to as a greyscale image. By convention 0 is black and 255 white, as shown in the figure below. The grey levels are the numbers in between, for example, in a linear scale 127 would be a 50% grey level.

24-bit RGB

RGB is the acronym for the colors Red, Green, and Blue. 24-bit RGB uses 8-bits of storage for <u>each</u> red, green, and blue component. In each component the value of 0 refers to no contribution of that color, 255 refers to fully saturated contribution of that color. Since each component has 256 different states there are a total of 16777216 possible colors (2^{24}).

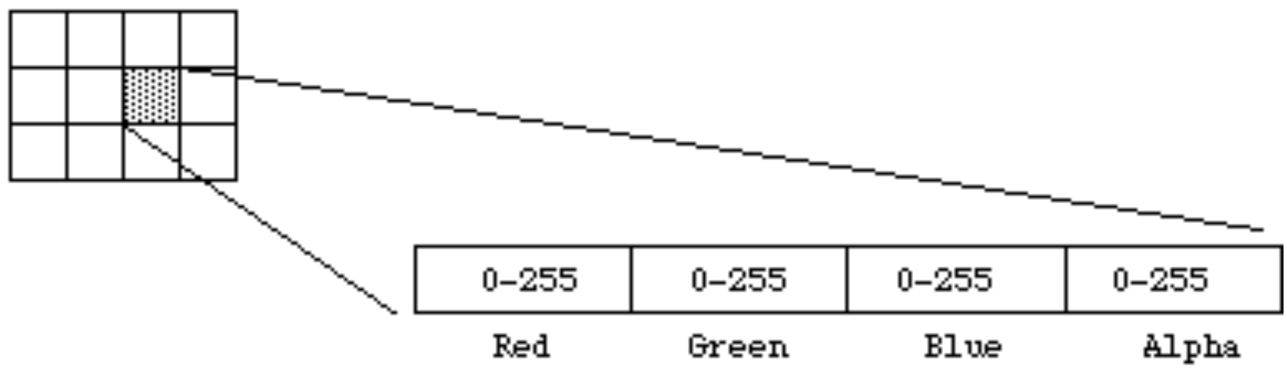


The idea of RGB color space is a fundamental concept in computer graphics. In RGB space any color is represented as a point inside a color cube with orthogonal axes r,g,b.



32-bit RGB

32-bit RGB is essentially the same as 24-bit color but with an extra 8-bit bitmap known as an alpha channel. The alpha channel is used to represent transparency.



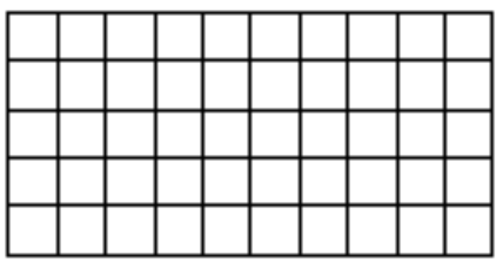
16-bit RGB

16-bit RGB is generally a direct system with 5 bits per color component and a 1-bit alpha channel.

Resolution

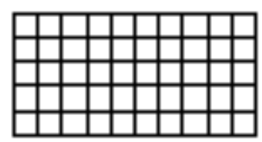
Resolution is an attribute of a bitmap that is necessary when visually viewing or printing bitmaps because pixels by themselves have no explicit dimensions. Resolution is specified as the number of pixels per inch PPI. The acronym DPI is Dots Per Inch. These two acronyms are often used interchangeably although they do have different meanings. PPI (Pixels Per Inch) refers display resolution, or how many individual pixels are displayed in one inch of a digital image. DPI (Dots Per Inch) refers to printer resolution, or the number of dots of ink on a printed image.

As an example, consider one bitmap which is 200 pixels horizontally and 100 pixels vertically. If this bitmap was printed at 100DPI then it would measure 2 inches by 1 inch. However, if the same bitmap was printed at 200 DPI, then it would only measure 1 inch by half an inch.



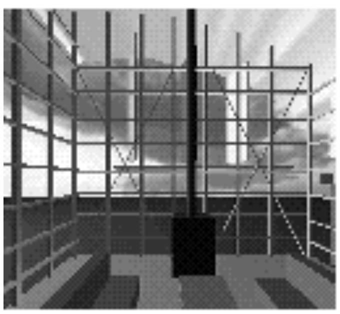
200 pixels by 100 pixels at 100DPI

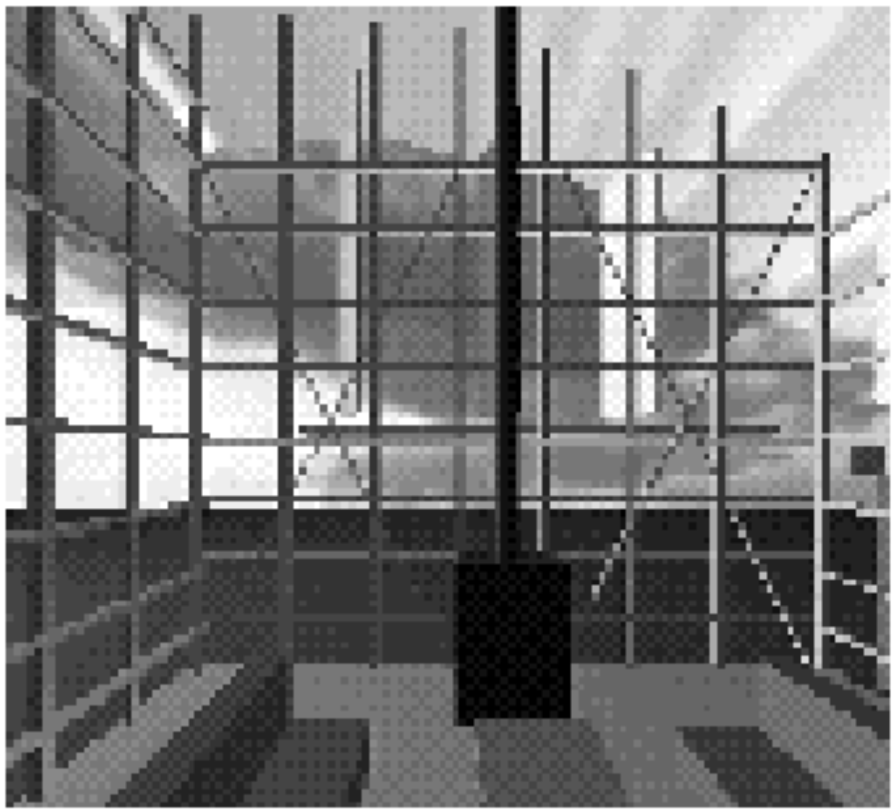
200 pixels by 100 pixels at 200DPI



Whenever a bitmap is displayed on a computer monitor resolution needs to be considered. Monitor resolution is the number of pixels a monitor can display. The higher the number of pixels, the sharper the image quality. Computer monitors have a range of resolution. As with printed matter the higher the resolution the less apparent the pixel nature of the bitmap will be.

For example, the following two images have identical in information content. However, they also have different resolutions, hence different pixel sizes. The smaller is 80DPI and the larger is 30DPI. The pixels are much more evident in the larger version.





Bitmap Storage

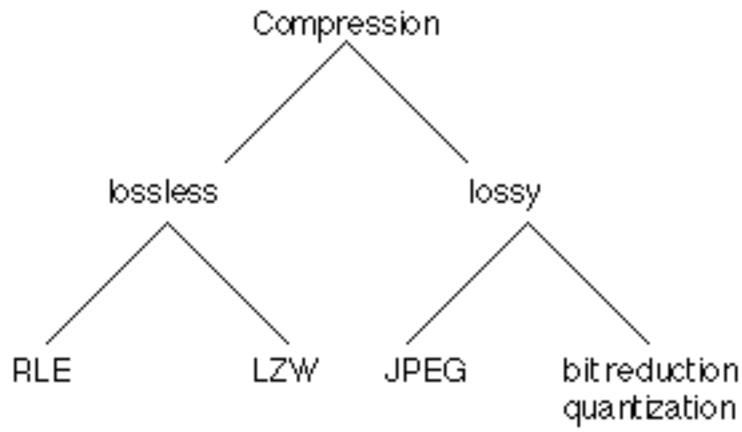
The most straightforward way of storing a bitmap is simply to list the bitmap information, byte after byte, row by row. Files stored by this method are often called RAW files. The amount of disk storage required for any bitmap is easy to calculate given the bitmap dimensions (N x M) and color depth in bits (B). The formula for the file size in Kbytes is

$$\text{size (KB)} = \frac{N * M * B}{8 * 1024}$$

where N and M are the number of horizontal and vertical pixels, B is the number of bits per pixel. The following table shows the file sizes of a few bitmap types if they are stored in RAW format.

image dimensions	color depth	file size
128 x 128	1 bit	2 KB
	8 bits	16 KB
	24 bits	48 KB
256 x 256	1 bit	8 KB
	8 bits	64 KB
	24 bits	192 KB
1K x 1K	1 bit	128 KB
	8 bits	1 MB
	24 bits	3 MB

As shown in the table, 24-bit images result in much large files, which is why compression is important. There are many file formats used for storing compressed bitmaps from the trivial to the very complicated. The complicated formats exist because of the very large bitmap files that would exist if compression was not used. There are two broad categories of compressed file format, those which are lossless (retain the bitmaps perfectly) and those which are lossy. The following shows the main hierarchy of compression techniques.



The crudest way of reducing the size of bitmap files is to reduce the color information, this is called bit reduction or quantization. For example, one could convert 24-bit bitmaps to 8-bit indexed bitmaps using dithering to simulate the lost colors. The most common lossy format by far is JPEG, a description of how it works is well outside the scope of this course. Its main advantage is that it can offer vastly better compression ratios than the lossless formats.

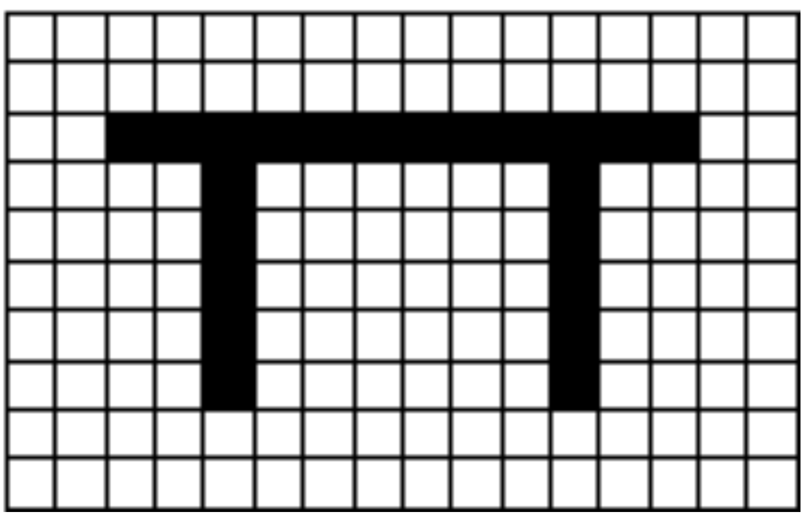
For example, consider the following bitmap the original of which is 500 x 350 pixels at 24-bit color. Using the formula given earlier the uncompressed file size is 500 x 350 x 24 / 8 / 1024 = 513K

Saved in greyscale (bit depth reduction) the file is 171K (3 times smaller), saved and compressed using an RLE compression algorithm is 388K (75% of the original). When the same images is saved using the LZW compression algorithm it is 188K (36% of the original). When the image is saved as JPEG it is 30K (a compression ratio of 17:1).

Lossless Compression technique RLE

The following is a description of the simplest lossless compression technique called Run Length Encoding (RLE algorithm) that is used with good effect for bitmaps with only a few colors.

Consider the following small, 17 x 10 pixel, 8-bit image.



If this was to be stored in RAW form, it would require 16 bytes per row for all 10 rows. However, the first two rows are all the same level, so it is more efficient to simply save the number of same colors in a run along with the run color. The first two rows instead of needing 16 bytes only need 2 bytes each.

In raw format the first three rows would be

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0
```

Using run length encoding the first three rows would be

```
16 0
16 0
2 0 12 1 2 0
```

While there are more details involved in actual implementations of RLE than described here this is the basic principle behind run length encoding. In order for RLE to achieve some degree of compression there needs to be runs of the same color, for this reason it is unlikely to be useful for highly colored images such as 24-bit photographs.