

Explanation of Image File Types

I often get asked about image file types – which formats are best for which applications. So, I figured I'd type up something to give a quick explanation of how graphics are handled on computers, and how the different file types actually store their information.

Note: this file was created in MS Word, and then converted to a pdf. In the process, the images were changed slightly, causing the anti-aliased images to not look as good (don't worry if you don't know what 'anti-aliased' means – it's described below). However, if you change the zoom level to 87% in Adobe Reader, the images look fairly close to the originals (at least on my computer).

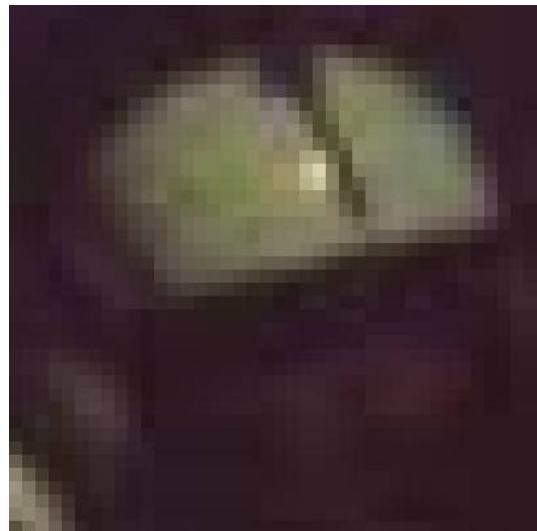
General Concepts

Pixels & Raster Image Formats

An image on a computer screen is made up of many tiny points called pixels. Pixels on screens are small enough that you can just barely see them. For example, in the photo of the aircraft to the right, it's difficult to see all the pixels that make up the picture.

But, if we blow up just the pilot's head, as in the picture below the aircraft, those pixels become quite obvious. If you held a magnifying glass up to the screen, it would look just like the blown up picture to the right.¹

Since graphics are displayed this way on the screen, it makes sense for files to store them that way, as well. And, in fact, that's the way most image types that people are familiar with work. When images are stored this way as an array of pixels, it's known as a raster type image, or sometimes as a bitmap type image. Images stored this way include bmp's, gif's (pronounced like the peanut butter), jpg's, tga's, png's, the photos from your digital camera, and probably just about every other image type you're familiar with.



¹ Sort of. As described in the Colors section on the following page, in actuality, monitors use red, green, and blue color elements to create the image. Three different elements are required for each pixel. So, when you look at a magnified image from an actual monitor, you'll see red, green and blue elements.

Once you zoom in on a picture or blow it up enough to where you can see the pixels, we say that the image is pixellated. Obviously, the more pixels you have making up your image (i.e. the higher the resolution), the more you can blow it up before the pixels become obvious. That's why camera manufacturers make such a big deal about the camera's resolution. In practice, if you can print at at least 150 pixels per inch (i.e. 22,500 pixels per square inch), your image will appear decent. 300 pixels per inch is even better. You can get away with lower resolution for photos than for line art.

Vector Image Formats

Not all images are stored as rasters, however. Some are stored as vectors. As an example, consider that you had a simple image that was a line from one point to another point. To store it as a raster each pixel would have to be specified between the points (as well as however wide the line is). In a vector format, the line would be stored as simply the coordinates of the two end points, along with another number for the width. Similarly, a circle would be stored as the coordinates for the center and another number for the radius. Probably the image format you're most likely to see that uses vectors is a wmf file. CAD files are also vector based.

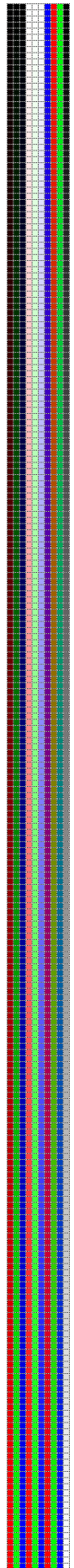
To be displayed, a vector format still has to be converted to pixels to show up on the screen. But, a vector format has huge advantages to scalability. It doesn't matter how much you zoom in on a certain portion of the image, or how much you blow it up. The computer 'knows' the shape is not just a collection of pixels, but is supposed to be a line, or circle, or whatever you're looking at, and so it will always regenerate the image as nice and smooth. Obviously, this format works best for line type images, and not so well for photos. The other disadvantage is that different programs (like Internet browsers) don't always support these formats, or don't support them well.

Colors

When you get down to the smallest elements on your screen that actually make the light that goes to your eyes to produce the image, those elements actually only make 3 different colors of light – red, green, and blue. However, by controlling the brightness of each of those three colors independently, it appears to our eyes that the screen is creating an entire rainbow of colors.

The way computers work with bits and bytes, it works out on almost all monitors that they can display 256 shades of red, 256 shades of green, and 256 shades of blue. 256 shades may not sound like a huge number, but it works pretty well. If you look at the image on the right, the first 3 columns show those 256 shades of each color (the rest of the columns in that image show different gradients that only change by one shade per row.) Unless you have superhumanly sensitive eyes, you can't really discriminate the difference from one shade to the next. And if you consider what happens when you mix all those different shades, 256x256x256 is over 16.7 million colors.

Just because most monitors work that way, doesn't mean all hardware works that way. Some applications use more than just 256 shades of color. Some use more information than just color. Some use different colors than red, green, and blue. Printers, for example, typically use cyan, magenta, yellow, and black inks to create images. Still, using 16.7 million colors is pretty typical for most common imaging applications.



Compression

If you tried to think of how a file might store all the information in a raster format, you might think that the easiest thing to do would be to specify how many rows and columns the image had, and then simply list the shades of red, green, and blue for each pixel. This is actually what some of the simpler file formats do, such as bmp's. The problem is that once you start getting a lot of pixels in your image, these files can get pretty big. If you could devise a way to store the image without individually specifying the color for each pixel, you could make the image a lot smaller. As an example, consider if you had a solid white image that was 100 pixels wide by 100 pixels tall. The simple way of storing the image mentioned above would require you to specify the resolution, and then define the red, green, and blue intensity for 10,000 pixels. That's 30,002 pieces of information (width, height, 10,000x3). If you had some sort of compression method, you could specify the resolution, define the color once, and then tell the computer to repeat that color 10,000 times. That's only 6 pieces of information (width, height, red intensity, green intensity, blue intensity, number of times to repeat). The first format requires 5,000 times more information than the compressed format. Obviously, that's a bit of an extreme example, actual compression is a bit more complicated to make it generalized for any image, and there are other methods of doing compression, but it illustrates why compression is important.

Anti-Aliasing

Anti-aliasing is a concept that's not really required to understand file types, but will help a bit in later sections. Look at the two circles to the right. The circle on the left appears a bit jagged, while the circle on the right appears smooth. Just below the circles is a blow up of a small area of the picture. It's obvious in the blow up why the circle on the right is smoother. Instead of transitioning straight from a black circle to a white background, shades of gray are used to smooth out that transition. Anti-aliasing is extremely useful in graphics, especially line art.

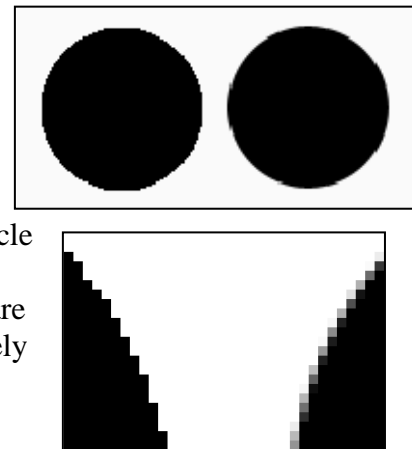


Image Formats

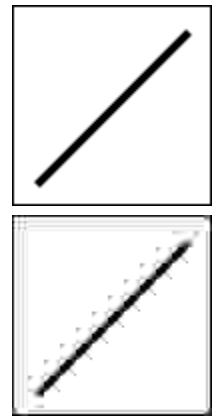
There are literally hundreds of image formats, but the ones described below are the ones you're most likely to come across.

JPG Files

Jpg's are the most common image file type. It's probably how your digital camera stores photos, and it's the format of most images on the web. The compression format jpg's use is nothing at all like the simple method I described. It makes use of Fourier transforms to encode the image, so a detailed explanation of how it works would be too long to include here. But, one thing that jpg compression takes advantage of is the fact that humans are more sensitive to changes in light intensity than changes in hue. So, by sacrificing a bit of information on hue, images can be made much smaller. Also, because of the way the

compression algorithm was defined, the user can specify just how much quality they're willing to sacrifice to reduce image size.

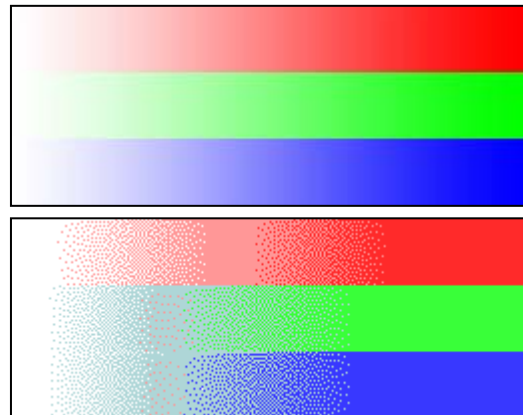
One disadvantage of jpg's is that they don't handle sharp lines very well – they create 'noise' around the line. If you don't do too much compression, and you have a background that can hide it, this noise isn't a big deal. But if you have line art, it can become quite apparent. As an extreme example, compare the two images to the right. The first is not a jpg, so has no noise. The second is a jpg with the max possible compression, showing the most possible noise. Obviously, you would never use such a high compression for real applications, but this illustrates the problem.



This points to another problem with jpg's – the regenerated image after compression is slightly different than the original. In other words, the jpg compression algorithm loses a bit of information about the image each time you save with it. If a compression format loses information in this way, it's known as lossy. This isn't a big deal if you never edit an image, or only edit it a few times, but it can be an issue for an image that's going to be edited a lot, or for an image where extreme accuracy is important. For those cases, many people will use a master copy in a format that doesn't lose information (but usually tends to be much larger), and then save a copy as a jpg when they're ready to share it.

GIF files

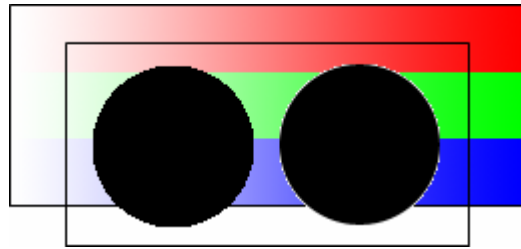
Gif files have been around a while. They're not used as commonly anymore, but they're still the best option for line art when you can't use a vector format, and you want to be sure that whoever you're creating the file for will be able to read it.



Gifs use a compression similar to the simple method described in the 'Compression' section above. However, gif files first define the colors that will be used in the image in a 'palette', and then when the color is needed later in the file, it's specified as a single number representing its place in the palette. The palette can contain up to 256 different colors. Those colors can be any of the 16.7 million colors possible, but each gif file is still limited to 256 colors.

The compression format gif's use doesn't lose any information when you save the file, which is known as lossless compression. Gif files also don't introduce any noise. However, because of the 256 color limit, photos don't turn out so well. If you have an image with more than 256 colors and save it as a gif, many editing programs will use dithering to try to fool your eye, but photos still don't look as good as when saved in other formats. Compare the two images to the right. The lower image was saved with a palette of only 8 colors to make the dithering more obvious.

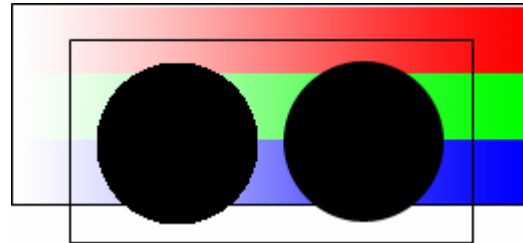
One interesting aspect of gif's is that they allow one of the palette colors to be specified as transparent. But – only one color can be transparent. The image to the right was originally created with two images. In the back is the gradient image from before, and in the front is the image of the circles from before. The background of the circle image was transparent. Remember that the circle on the left wasn't anti-aliased, while the circle on the right was. The anti-aliased circle looks great on a white background, but the anti-aliasing stands out like a sore thumb on a dark background because it was done using shades of grey. If you use transparency in gif's, you have to keep this in mind.



PNG Files

Png files are basically a replacement for gif files. They've been around since 1995. That makes them old enough that most software now supports them, but they're still not quite as widely supported as gifs. Pngs use a compression method similar to gif's, but are not limited to a 256 color palette. Png images can contain the full 16.7 million colors available.

Png images also support transparency, but in a more refined way than gifs. Instead of simply specifying one palette color as transparent, png files specify the transparency of every pixel. In other words, they don't just specify red, green, and blue – they specify red, green, blue, and transparency. The advantage is that when a transparent png file is overlaid on top of an image, you can still use anti-aliasing. However, instead of specifying the anti-aliasing as shades of the color, it's specified as varying transparency. The image above illustrates this. This time, the circles were saved as a png file. You can see that the anti-aliased circle now looks very smooth, with no outline around it.



TIFF Files

Tiffs can contain a mix of raster or vector graphics. Tiffs can also be compressed in different ways, using the lossy method that jpgs use, or lossless methods. Though typically, tiffs use lossless compression for raster images. Tiffs became popular as desktop scanners became popular because of their ability to use 16.7 million colors with a lossless compression (before pngs when jpgs and gifs were the dominant formats). They are still fairly widely used because they can actually store images with more than the 16.7 million colors, or represented by different primary colors than red, green, and blue (e.g. cyan, magenta, yellow, black). Tiff files tend to be used more for professional applications, and aren't as widely supported as other formats. In particular, most web browsers won't display tiffs.

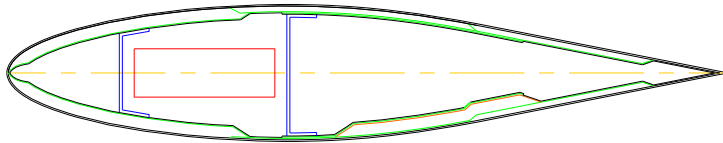
BMP Files

Bitmaps are a very simple file format, not really recommended for any use, but you may come across them from other sources. They allow pictures to use the full 16.7 million colors. Although they can be compressed, they typically aren't, and most image editing programs don't even offer the option to compress them. Obviously, the image quality will be very good, but the files will also be quite large.

WMF Files

Wmf files can actually contain a mix of raster and vector information, and even text. They're not very widely supported at all. Typically, I only use them to transfer an image from CAD into a Microsoft Office program (Word or PowerPoint). The image to the right is an example of a wmf.

No matter how much you zoom in on the picture or blow it up for print, the curves will always appear smooth.



Summary of the Most Common File Types

Extension	Advantages	Disadvantages	Recommended for
Jpg	High compression ratios possible, Widely supported	Lossy Compression, Noise around edges	Photos, especially for web
Gif	Lossless compression, Widely supported	Limited to 256 colors per image	Line art & transparency, but only when png not feasible
Png	Lossless compression, 16.7 milion colors	Not as high of compression as jpgs for photos, not as widely supported as gif	Line art & transparency, Archiving Photo Data
Tiff	Multiple compression, multiple color options	Not as widely supported as other formats	Scans, Archiving Photo Data
Bmp	Widely supported by Microsoft, Simple	Typically no compression	Nothing
Wmf	Vector graphics	Not widely supported	Transferring CAD dwgs into Word or PowerPoint