LINUX JOURNAL
24 YEARS OF LINUX JOURNAL (1994–2017)

Graphics Formats for Linux

Gerald Graef

Issue #23, March 1996

Gerald Graefs provides a comprehensive guide through the jumble of graphics formats currently available.

If there is a single unifying feature of the computer world, it is non-conformity. In no place is this better illustrated than computer graphics. Literally dozens of different formats exist to do the same thing: store graphic images. Proprietary formats, independent free standards, and international standards all vie for attention. Fortunately, the computer community has settled on just a handful of formats. Some, like TIFF, are the jacks-of-all-trades and are widely used. Others, like GIF, have a niche that they dominate. But whatever the context, there is always more than one way to store images.

This article first provides a brief introduction to graphics on Linux, including format and type, compression, file structure, and various features. It then continues with descriptions of the common graphics formats found in the Linux world and programs with which to manipulate them.

Format

Graphics formats come in two basic varieties: **raster** and **vector**. Raster formats store the entire image pixel by pixel and are the most commonly-used formats. Vector formats, also called descriptive formats, do not contain pixel image data; instead, they contain a series of commands telling a display program how to draw the image.

The most common use of vector formats is in computer aided drafting (CAD). Describing a line, for example, as a starting point, an ending point, and a color, as opposed to many colored points in a field of points, allows a program to manipulate the line as an object, making it easy to change color, length, and position. Another common use is in raytracing programs. These programs take geometric objects and trace the path of light rays to find shadows and highlights. Since the rendered objects are geometric shapes, it is much easier to store an image as "dodecahedron at (3,4,3) with size 2 and color 27" than as an array of pixel values, especially since the scene to be rendered is three-dimensional. Of course, once the scene is raytraced, the resulting image is stored in a raster format.

Type

Images come in four basic types: monochrome (black and white), grayscale, color palette, and true color.

Grayscale and color palette images typically have 2, 4, or 8 bits per pixel. They use lookup tables that must be included in the image file; each entry in the color lookup table gives the red, green, and blue values for a color. Grayscale images have a similar lookup table for gray values. Each color or gray value is in a range of 0-3 for 2-bit images, 0-15 for 4-bit images, and 0-255 for 8-bit images. The actual image pixel data is a series of color or gray values corresponding to table entries which give the actual color of each pixel.

True color images are 24-bit images and are usually stored as 8 bits of red, 8 bits of green, and 8 bits of blue per pixel. Some formats may use different color models than RGB. The JPEG format, for example, uses a colorspace based on the luminance and two chrominance values (YCbCr). True color images do not need a color lookup table or palette.

Compression

Compression is integral to most graphics formats. An uncompressed 640x400x256 color image requires just over 256,000 bytes, while a true color image of the same size requires three-quarters of a megabyte. Although many compression schemes exist, only a few are common in graphics applications. LZW, LZ77, Huffman, and run-length encoding are lossless, meaning that the image is reproduced exactly when uncompressed. The cosine transform is the basis for the “lossy” JPEG format. We aren't concerned with the details of the compression schemes here.

JPEG gives by far the best compression, but at the expense of exact image reproduction and speed. Acceptable results are common with compression ratios of 25 to 1, that is, the compressed file is 25 times smaller than the original. Ratios of 100 to 1 may be possible if image quality is not a primary concern.

[FOOTNOTE:A better algorithm does exist, however: fractal image compression. This method gives good results at 100 to 1 and is finding many applications, though it has not yet been widely implemented as a general purpose graphics format.]

LZW and LZ77 are the best lossless schemes and produce similar results: usually around 2 to 1 compression. The modified LZ77 used in the PNG graphics format is slightly slower at compressing images than LZW, but it creates smaller files and is faster at decompressing. LZ77 is also used by the general purpose compression programs ZIP, GZIP, and PKZIP. Until recently, the use of LZW was much more common than LZ77. Its use is falling, however, because it is a patented algorithm, and UNISYS requires royalties for its use. Nonetheless, both GIF and TIFF, two of the three most common formats, use it.

Run length encoding (RLE) is the most common scheme and the least effective. It is, however, very fast compared to other compression algorithms. Its basic mechanism is very simple: when a string of pixels with the same value is encountered, RLE outputs the pixel value and the number of consecutive pixels with that value. Complex images do not compress well with this scheme, although monochrome images often compress very well.

Structure

Simple formats consist of nothing more than the image data and a short header giving the minimum amount of information necessary to display the image: size and number of colors. More robust formats often employ a tag or chunk structure. After an initial header, the image is stored in a series of chunks, each prefixed by a code telling the decoding software what the chunk contains. The biggest chunk is the actual image data. Other chunks may include comment blocks, palette information, or other special image information. Software may only support some chunks of a given format: if an unknown chunk is encountered, it is simply skipped.

Finally, three special features bear closer examination: magic numbers for format identification, gamma correction, and alpha transparency masks. In some cases it is possible to identify the format of a file by looking at the file itself. Many formats use magic numbers—special (unique, it is hoped!) codes—to identify the format. GIF files, for example, have ”GIF87a” or “GIF89a” at the start of the file. Because these magic numbers are ASCII coded, using **less**, **strings** or even **cat** (though that can accidentally change your character set) to display them on the screen is good enough for identifying them. For example, to look for GIF

Gamma correction is an important part of the proper display of an image. Whenever an image is displayed on a monitor, the monitor's characteristics affect the image. How the monitor handles varying degrees of brightness is especially important. In general, the brightness on the screen is related to the brightness levels of the original image by a simple formula: screen brightness equals image brightness raised to the gamma power. The value of gamma for most monitors is around 2.5. To compensate for this, image capture hardware and software may make an "opposite" operation on the image. The result is an image with a gamma of 0.45. When this image is displayed, the gamma of 0.45 and 2.5 cancel each other out, producing an apparent gamma of about 1 (called linear brightness scale).

But what does gamma mean visually? An image displayed with a gamma less than one uses more of its pixel codes in the darker areas, that is, darker areas have better color resolution. A gamma value greater than one uses more of its pixel codes in the lighter regions. A murky image, or an image with too much contrast, may need to have its gamma corrected. Correcting the gamma of an image is inherently lossy because of round-off error during the taking of powers; hence, gamma correction should be performed when the image is displayed, keeping the original image file intact without loss.

Alpha transparency masks are a way of hiding, or masking, parts of an image. In addition to image data, a value is included in the image file for every pixel. A value of 2bitdepth-1 is opaque and the pixel is displayed normally. A value of 0 is transparent, allowing the background color of the screen to show through—the image is not visible at all. The simplest application is to make a non-rectangular image. Define an array the size of the image and then draw a line between two opposite corners. Set the value of each point above the line to one, and the values below to zero. The result will be a triangular image.

The Formats

BMP

BMP is the native bitmapped format used by Microsoft Windows. It is a minimal format that has few features and uses simple run length encoding for data compression. With the widespread use of Windows, BMP is the most common format. In many ways BMP is very similar to the PCX format, and has assumed the role that PCX once held.

BMP encodes 1, 2, 4, 8, and 24 bit images. OS/2 uses a similar BMP format, the only difference being a slightly simpler header. The first two characters of the BMP header are always "BM".

WMF

Another format native to Microsoft Windows is the Windows MetaFile (WMF). WMF uses Windows' Graphical Device Interface (GDI) function calls to store images that appear repeatedly in applications. The GDI calls provide support for setting up the screen, defining regions, colors, text, pixels, lines, polygons and bitmaps. WMF supports uncompressed monochrome, color palette, and true color images.

GIF

The Graphics Interchange Format (pronounced "jiff") was originally developed by the Compuserve Information Service as a color replacement for the earlier RLE format. Since the 1987 release, GIF has become the standard for graphics interchange, especially on networks. A second release in 1989 added several new features to the format. GIF is a lossless format based on the LZW compression scheme. It uses a tag system to identify extension blocks in the file, although only a few tags have been defined. The biggest assets of the format are its relative simplicity, excellent compression, and widespread availability.

However, the format has two major drawbacks. First, it is based on a copyrighted compression scheme and commercial software using it must pay royalties to Unisys, the patent holder. Second, GIF can be used only for images with 256 or fewer colors. (Strictly speaking, that is. Since GIF supports local color maps, the palette can be changed in the middle of an image allowing for more than 256 colors. Unfortunately, much of the available software does not support, or supports poorly, this feature, and it is, at best, clumsy to use.) GIF files can be identified by the string **GIF87a** or **GIF89a** at the start of the file.

JPEG/JFIF

JPEG (pronounced "jay-peg") is a standard for lossy compression of images. JPEG achieves compression by breaking the image into 8x8 boxes of pixels, performing a mathematical operation called a cosine transform on each, and throwing out the high frequency/small detail components; the more components thrown out, the greater the compression and the poorer the image quality. The remaining frequency components are then run length encoded and compressed with the Huffman algorithm. To view an image, a JPEG file must be uncompressed, decoded, and an inverse cosine transform performed. This three-step process makes displaying JPEGs very slow. Fortunately, JPEG produces excellent compression with little or no visible image loss.

JPEG itself is not actually an image format but a set of standards for compressing image data. JFIF, the JPEG File Interchange Format, is the format commonly referred to as JPEG. JFIF does not support all of the features of JPEG, but is intended as a minimal implementation for image transfer. A full featured implementation of JPEG is included in version 6.0 of the TIFF format. Designers hope the two implementations of JPEG, one minimal and one full featured, will deter software vendors from defining proprietary formats based on JPEG (early Macintosh versions were especially notorious for incompatibilities).

JPEG works best on real world images; line drawings and cartoons will not compress nearly as well as scanned images. Unlike most other formats, JPEG does not store a pixel as red, green, and blue values. Instead it uses a format called YCbCr. Since most display hardware uses RGB values, the YCbCr values must be converted—yet another step slowing decompression. The JFIF header includes the ASCII characters "JFIF" for format identification.

PBM/PGM/PPM

These three formats are intermediate formats used by the PBMPLUS utilities. The acronyms stand for Portable BitMap/GrayMap/PixMap. PBM is for monochrome images, PGM for grayscale images with up to 256 shades of gray, and PPM for color images using up to 24-bits of true color. A fourth "format" is the Portable AnyMap, PNM. PNM is not actually a format itself. A program that uses PNM can read and write PBM, PGM, and PPM files. PNM is used for utility programs that support multiple image types. For instance, since the image type of a TIFF file may not be known, PNM reads the TIFF file and writes the appropriate file type.

Each of the four formats can read the other ones that carry less information. That is, a PGM utility reads PGM and PBM, a PPM utility reads PPM, PGM, and PBM. PBM, PGM, and PPM utilities always write in their own format, while PNM utilities generally write whatever format they have read. The formats store data either as ASCII or binary data and are otherwise basic formats consisting of a header and image data. The header consists of a magic number to identify the format, image size, and (except for PBM) the number of colors/gray shades. The magic strings are **PBM P1(P4)**, **PGM P2(P5)**, and **PPM P3(P6)**, where the first code is the code for ASCII data, and the code in parentheses is for binary data. True color images store pixel data as a triplet of numbers for RGB data. For more information on the PBMPLUS package, see the section on software below.

The PCX format is the native format of Z-Soft's PC Paintbrush program and uses run length encoding. As one of the first general purpose formats, its use has been very widespread: few programs exist that do not recognize it. PCX is a basic format consisting of a 128 byte header followed by image data. Monochrome as well as 4, 8, and 24-bit color images are supported. The palette for 4-bit images is included in the header while the 8-bit palette is appended after the image data. This non-uniformity is the result of an older format being updated for newer hardware. For this reason, the use of PCX has dwindled in favor of the more coherent and unified BMP.

PNG

Amazingly, with all the other formats available, there has been an important niche left unfilled: a portable, relatively simple, free standard for the lossless exchange of true color images—in short, a 24-bit version of GIF with a non-patented compression algorithm. There are existing formats that can be used, but only TIFF has the necessary features, and it suffers from over-completeness: very few implementations make use of the entire TIFF specification. What is needed is a format that is simple enough that any image saved under it can be read by any viewer supporting it.

Compuserve called their development specification GIF24, but when a group of Internet graphics experts developed PNG (Portable Network Graphics, pronounced “ping”), Compuserve adopted it as the successor to GIF. PNG is a natural extension to GIF, although it is not backwards compatible because of a change in compression scheme. In addition to the original GIF features, PNG supports true color images up to 48 bits and grayscale images to 16 bits, as well as full alpha-channel, gamma correction, and detection of file corruption. On 8 bit and larger data, PNG can use a preprocessor on the image data prior to compressing. In many cases this processing improves the compression efficiency and results in smaller file sizes. Expect to see PNG files appearing at an archive near you soon.

PS/EPS

PostScript (PS) is a page description language developed by Adobe Systems that is both a descriptive and raster format and has become one of the most common printer languages. PostScript files are created by many application programs as a device-independent output format. Encapsulated PostScript (EPS) is a limited version of postscript for single pictures and is used for images to be included, or “encapsulated”, in programs or postscript files. The first line of a postscript file is a line of the form:

```
%!PS-Adobe-3.0
```

where the number refers to the PostScript version the file was created under. EPS files append **EPSF-3.0** to this line. PostScript is a large and versatile language. Display PostScript is an implementation of PostScript for controlling video hardware and is used by NeXT computers and software.

TGA

The Targa Truevision graphics format was developed for use in Truevision's product line. TGA uses run length encoding to store grayscale, color table, and true color images, and can include comments, gamma, alpha, color corrections, and a “postage stamp” version of the image. TGA was one of the first true color formats and is still used by some applications like the Persistence of Vision Ray Tracer. TGA files may contain a block at the end of the file that includes the text **TRUEVISION-XFILE**.

TIFF

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.