Effective Variations on Opened GIF Format Images

Hamza A. Ali1[†] and Bashar M. Ne'ma2[††],

Isra Private University, Amman, JORDAN

Summary
The CompuServe GIF format and the LZW compression method used to compress image data in this format is investigated in this paper. Because of its better compression and greater color depth, JPEG has generally replaced GIF for photographic images. Thorough study and discussion on GIF format images is carried out in details in this work. Although, opening the header of GIF format images is difficult to achieve, we opened this format and studied all acceptable variations which may be have influence on the viewing of any GIF format images. To get appropriate results, all practical is carried out via both Object Oriented Programming (OOP) and concepts of software engineering tools.

Key words:
GIF format, Image Header, Image processing, LZW, Compression, Software Engineering Tools, and Security.

1. Introduction

There are a large number of formats for storing digital images. Most were developed for use with a particular program, but some of these formats have become de facto standards, and are used by many programs. A few formats were developed specifically for interchanging files between different programs and computers.

Most bitmap formats have provision for reducing the file size by 'compressing' the image data. The amount of compression depends on both the compression method and the nature of the image. Most compression methods work well on images that have large areas of uniform colors. For example, a simple line drawing could usually be compressed to less than 1/10th of its original size.

So many graphic file formats are available with different applications and importance. Basically, they can be grouped into two main categories [1, 2];

- ***Bitmap formats***: images created by individual pixels displayed in various colors, each pixel can hold up to 24 bits (3 bytes) of color information. Bitmapped formats have a fixed resolution and can not be resized without some effects on the clarity of the image. They are best suited for "photographic" quality images. Examples of bitmaps graphic file formats are; Graphic Interchange Format (GIF), Joint Photographic Expert Group (JPG or JPEG), Tagged Image File (TIF), Picture Format (Pict), Portable Network Graphic (PNG), Photoshop native file (PSD), PCX from Zsoft, Kodac PCD. Usually scanners and digital cameras acquire images in bitmapped formats.

- ***Vector formats***: images are a series of pixels that are "turned on" based on a mathematical formula. They are basically defined by shapes and lines with no more than 256 colors. They are not resolution dependent, infinitely scalable and not appropriate for photo realistic images. Examples of vector file formats are Windows Metafile Format (MWF), PostScript Format, Portable Document Format (PDF), and Computer Graphic Metafile (CGM).

In the rest of this section, a short definition is included for the most widely used computer graphic formats.

- **GIF** (CompuServe Graphics Interchange Format): It is an 8-bit-per-pixel bitmap image format that was introduced by CompuServe in 1987 which is suitable for file interchange due to its wide support and portability [3, 11]. A good compression method is built into the format.

- **TIF/TIFF** (Tagged Image File Format): A very flexible bitmap format, supporting mapped and unmapped images. It serves as a wrapper for different bit-stream encodings for bit-mapped (raster) images. TIFF is a container format for storing images, including photographs and line art. It is now under the ownership of Adobe. Originally created by the company Aldus [11, 6] for use with what was then called "desktop publishing," TIFF is a popular format for color and black and white images. The TIFF format is widely supported by image-manipulation applications, by publishing and page layout applications, by scanning, faxing, word processing, optical character recognition and other applications. Several compression options are available.

- **JPG/JPEG** (Joint Photographic Experts Group): A bitmap format for unmapped images, developed as a standard by ISO and CCITT. The main feature of this format is extremely good compression of continuous-tone images. Some information is lost in the compression, and there is a trade off between amount of

Manuscript received May 5, 2008
Manuscript revised May 20, 2008

compression [5] (which may be specified by the user) and image quality. For most images, 20:1 compression may be achieved without noticeable degradation. (By comparison, GIF format, and TIFF format with LZW compression, give about 2:1 compression with continuous-tone images. Because GIF images are mapped, they effectively have an additional compression of 3:1 for color images, but information is lost in the color reduction).

- **PCD** (PhotoCD): A bitmap format used by Kodak for their Photo CD disks. Each file contains 5 or 6 versions of the image, at different resolutions. PCD stands for Photo CD and means a high (actually the highest) resolution format for images on a CD. It was developed by Kodak. A PCD file contains five different resolution (ranging from low to high) of a slide or film negative. Due to it PCD is often used by many photographers and graphics professionals for high-end printed applications [6]. PCD files can be accessed in a variety of dimensions and color depths, which is quite handy, and the format is good at encoding and storing authentic color information. The PCD files are typically 4 to 6 MB in size and, therefore, take some time to transfer via the web [7].

- **PCX**: A mapped bitmap format developed by Zsoft for their paint programs. In its early days the format only supported 4 or 16 colors. But later versions support 256 colors and have the ability to store 24 bit color images [5, 8].

- **CGM** (Computer Graphics Metafile): A vector format developed for file interchange. It is the most common DOS-based format. It can support very elaborate graphics. It is originally designed as a graphical file exchange and printing format for CAD systems, then it is enhanced with advanced primitives such as splines and polybezier curves, & tiled raster for image data. Latter versions Added application structuring so that a CGM file can contain non-graphical information such as part numbers, descriptions, signatures, notes, etc. This gives CGM files intelligence, so that you can drill into a graphic for more detailed information. CGM is ideal tool for web publications [9].

- **WMF** (**W**indows **M**etafile **F**ormat): Designed for easy exchange between Windows programs via clipboard. It is a common clipart format.

- **P**ost **S**cript Format: Developed by Adobe Systems for typefaces. Expanded for many graphic elements. **E**ncapsulated **P**ost **S**cript (.eps) is another version of this format.

- **PDF** format (Portable Document Format): It is an Adobe Acrobat file format developed to integrate color, graphics and text together in a searchable format.

2. GIF Format Features

In 1987 CompuServe Published the first GIF specification called GIF87a [3]. This specification was freely distributed, and the format was adopted by practically every image processing application. Until recently, CompuServe Graphics Interchange Format (GIF) was the most widely used format for image storage.

CompuServe later released an enhanced, upwardly compatible version of standard known as GIF89a. It has feature of storing multiple images in one file, accompanied by control data. It is used extensively on the web to produce simple animations. The optional interlacing feature stores image scan lines out of order in such a fashion that even a partially downloaded image was somewhat recognizable. This has helped GIF's popularity also, as a user could abort the download if it was not what was required.

GIF format uses a palette of up to 256 distinct colors from the 24-bit RGB color space. It also supports animations and allows a separate palette of 256 colors for each frame. The color limitation makes the GIF format unsuitable for reproducing color photographs and other images with continuous color, but it is well-suited for more simple images such as graphics or logos with solid areas of color. GIF images are compressed using the Lempel-Ziv-Welch (LZW) lossless data compression technique to reduce the file size without degrading the visual quality.

2.1 GIF usage

- GIFs are suitable for sharp-edged line art (such as logos) with a limited number of colors. This takes advantage of the format's lossless compression which preserves very sharp edges (in contrast to JPEG).
- GIFs can also be used to store low-color sprite data for games.
- GIFs are used for small animations and low-resolution film clips
- In view of the limitation on the GIF formation to 256 colors, JPEG is a more commonly used format for digital photographs. JPEGs can save information on more than 16 million different colors and use more aggressive lossy compression which has a less noticeable effect on photographs than it does on images with sharp edges.

- In circumstances where speed is more important than reduced file size, uncompressed bitmap formats such as Windows bitmap are more commonly used than the GIF format, since uncompressed bitmaps contain raw pixel information and can be displayed very quickly.

2.2 Main features:

The main features of GIF images can be summarized as: (1) They consist of up to 256 colors using 1 to 8 bits per pixel, and (2) Multiple Images per File.

Because of its better compression and greater color depth, JPEG has generally replaced GIF for photographic images. However, GIF continues to be used for other applications, but legal entanglements have certainly condemned it to obsolescence [10 - 12].

2.3 Byte Ordering:

The GIF format stores multi-byte integers with the least significant byte first (Little-endian). Bit strings are read from the least significant bit to the most significant bit. In bit strings that cross byte boundaries, the bits in the second byte are more significant than the bits in the first byte.

3. Classification of GIF File:

A GIF file consists of a fixed area at the start of the file, followed by a variable number of blocks and ends with an image trailer. In the GIF87a format the variable area consists solely of image definitions. In the GIF89a format these can be either images or extension blocks. Figure 1 below illustrates a block diagram for the general format of a GIF file structure.

3.1 File structure:

This GIF file structure consists of the followings:

***a- GIF Header*:**

The GIF header is required and must occur at the very start of the file. This *header* allows the application to identify the format as GIF and determines its version. It consists of 6 bytes and from practical work we find it as shown in table 1.

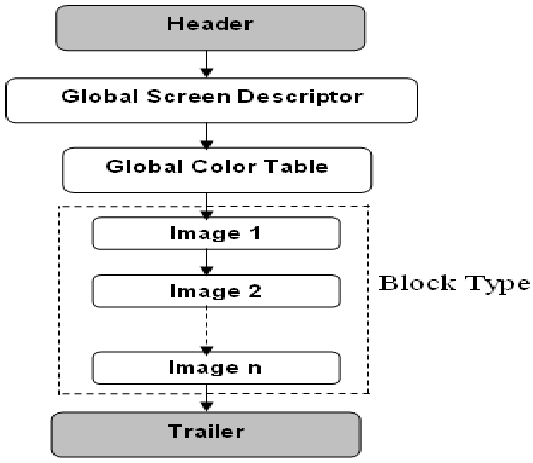


Figure 1: GIF File Structure

Table 1: GIF header

Name	Size	Comments
Signature	3 bytes	Must be ASCII String GIF.
Version	3 bytes	Must be ASCII string 87a or 89a.
Note: For version 89a there are two extra byte which may be allocated for inserting a special characters refers to the compression method may be used by any person who can compress the GIF image in another methods.		

b- Global Screen Descriptor:

The *global screen* description defines the logical screen area in which the individual images in the GIF file are displayed. The screen descriptor structure consists of 7 Bytes. Practical extraction of the content can be listed as shown in table 2.

c- Global Color Table:

The individual images within the file can either use the global color table (GCT) or define a color table of its own. Having the images in a file sharing the global color table reduces the file size and makes it easier for systems that can display a limited number of colors. In addition, the global color table specifies the background color for the logical screen.

Table 2: Global screen descriptor

Code	size	Comments
Screen Width	2 bytes	
Screen Height	2 bytes	
Bit Fields	1 Byte	
Global Color Table size (GCT)	Bits 0-2	Gives No. of entries in GCT.
Color Table Sort Flag	Bit 3	Set if Colors Ordered on Importance.
Bits Per Pixel	Bits 4-6	Bits Per Pixel -1.
Global Color Table Flag	Bit 7	Set when there is color table flag.
Background color	1 byte	Index into GCT.
Pixel Aspect Ratio	1 byte	If this value! =0 W and H not equal.

d- Block Types:

After the global color table, the variable part of the GIF file begins. The file contains a sequence of blocks that are identified by a 1-byte code at the start of the block. Table 3 lists the block types and the associated block code.

Table 3: Block code

Code	Type	Comments
21 (H)	Extension.	Only in GIF89a more image specification
2C (H)	Image block	Defines an image within the GIF file.
3B (H)	GIF terminator.	Marks the end of GIF file.

e- Trailer:

After the block code, the GIF file is terminated by the trailer which is mostly used to include the information or signature of the authors.

3.2 Interlacing:

In general, GIF images are stored from top to bottom, left to right. However the interlacing flag in the image header is set, the rows of pixel data are not transmitted in strict top-to-bottom order. The interlacing process serves the same function as progressive JPEG.

4. Compressed Data Format

The pixel data within a GIF image is compressed using a process known as Lempel-Ziv-Welch (LZW). LZW is what is known as *dictionary-based compression scheme*. By that, we mean a compression method that maintains a list or dictionary of sequences within the uncompressed data. During compression, when these sequences occur in uncompressed data they are replaced by a code that references the sequence in the dictionary.

The *trick* in dictionary-based compression is how to transmit the dictionary in the compressed data. The most common dictionary-based compression schemes in use are based upon those described by Abraham Lempel and Jacob Ziv (1977 and 1978), and known as LZ77 and LZ78, respectively [13 - 15]. LZ77 uses a sliding window into uncompressed data to implement the dictionary. LZ78 builds a dictionary dynamically from the uncompressed data.

4.1 GIF Compression:

LZW is a variant of the LZ78 process that was described in a paper by Terry Welsh of Sperry (now Unisys) in 1984. CompuServe adopted it for use in the GIF format shortly afterwards. In the LZW method, the compressed data stream consists entirely of code that identifies strings in a dictionary. The dictionary is initialized so that it contains each possible data value as a predefined string. For example, if 8-bits data is being encoded, the dictionary initially contains 256 1-byte strings containing the vales 0-255.

The compression reads characters from the input stream and appends them to the current string until the current string no longer has a match in the dictionary. At that point it outputs the code for the longest matching string adds the no matching string to the dictionary and finally starts a new string that contains the first no matching character.

Practical **Algorithm (1)** illustrates how the dictionary is created in the LZW process. Here we are assuming that **8-bits** data is being used and that the function output writes **9-bits** codes to the output stream.

4.2 GIF Decompression

An LZW Decompresser reads one code at time from the compressed stream while maintaining the dictionary in the same way the compressor dose. Each code simply gets translated from the dictionary.

The only subtlety in decompression is that it is possible for the compressed stream to contain codes that have not been defined in the dictionary. Algorithm (2) illustrates the LZW expansion process.

Algorithm (1): Simplified LZW Compression:

```
Global string DICTIONARY [0 .. 511]
Global NEXTCODE = 256
Procedure Initialize
Begin
        For I=0 to NEXTCODE -1 Do
        DICTIONARY [I] = CHARACTER (I)
End
Function SearchDictionary (String SEARCH)
Begin
        For I=0 to NEXTCODE -1 Do
                                        Begin
                          If DICTIONARY [I] = SEARCH Then
                          RETURN I
                       End
         RETRUN -1
End
Procedure Compress (String DATA)
Begin
        Initialize;
        LASTSTRING=NULL;
        For I= 1 to Length (DATA) Do
                    Begin
                       Current String=LASTSTRING + DATA [I];
                       CODE= Search Dictionary (CurrentString)
                       IF CODE<0 Then
                       Begin
                          CODE= SearchDictionary
(LASTSTRING);
                          OUTPUT (CODE)
                       End
                       Else
                       Begin
                            LASTSTRING= CURRENTSTRING;
                       End
                End
        Output (SearchDictionary (LASTSTRING))
End.
```

Algorithm (2): Simplified LZW Expansion:

```
Procedure Expand
Begin
   LASTCODE = InputCode ()
   Output (LASTCODE)
   While NOT EndOfStream Do
       Begin
          CODE = InputCode ()
          If CODE < NEXTCODE Then
          Begin
             Output (Dictionary [CODE])
             Dictionary[NEXTCODE]=Dictionary
           [LASTCODE]    +Dictionary [NEXTCODE-1] [1]
             NEXTCODE=NEXTCODE+1
                  LASTCODE=CODE
          End
       Else
         Begin
              Dictionary    [NEXTCODE]    =    Dictionary
           [LASTCODE]
                        +Dictionary [LASTCODE] [1]
           NEXTCODE=NEXTCODE+1
           Output (Dictionary [CODE])
           LASTCODE=CODE
         End
       End
End
```

5. Conclusions and Results:

The GIF format was the first image format to be universally accepted. Unfortunately, legal problems have ended GIF development. Unlike other major graphics formats, no enhancements to GIF are under way. This coupled with inherent limitations compared to other formats, has made it essentially dead.

We discussed GIF in details and as a practical work, we are open the header of GIF image format which may be useful to use and applicant in the following major applications:

- Make statistical evaluation for GIF images.
- Use GIF Images as a STEGO file in
- steganography applications.
- Insert secure image in GIF list of images used in Visual Cryptography.
- Create an Animated Images useful in Simple Game building.

References

[1] Mark M., "Native file format: Graphic File Formats", *Dept. of Technology & Cognition, University of North Texas*, 2002.

[2] Mark M. ," BMP, CGM, PCX, and More Windows Bitmap and Vector Graphics", *Dept. of Technology & Cognition, University of North Texas*,2004.

[3] *CompuServe Inc.*, "Graphics Interchange Format (tm): A standard defining a mechanism for the storage and transmission of raster-based graphics information", June 15, (1987).

[4] *CompuServe, Inc*, "Graphics Interchange Formats GIF Specification", CompuServe, Columbus, OH, (1989).

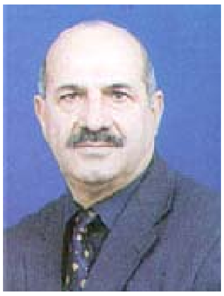
[5] John Miano, "Compression Image File Format", *Addison-Wesley*, 1999.

[6] Peter Siegel and Nina Grigoryeva,"Using Primary Data to Design Web Sites for Public and Scientific Audiences", *American Museum of Natural History*, USA.

[7] Kentaro Toyama, Ron Logan, and Asta Roseway ,"Geographic Location Tags on Digital Images", *Microsoft Research*, Microsoft Way,Redmond WA, 98052, 2003.

[8] Ralf Steinmetz, Klara Nahrstedt, "Multimedia Systems", Springer, 2004.

[9] Dieter Weidenbrück, "Manipulation of properties of CGM objects in viewing environments", *Conference proceedings of XML* 1999, Philadelphia.
[10] Brown, C. and Shepherd J., "Graphics File Formats", *Manning, Greenwich*, CT, (1995).
[11] John Miano, "Compressed Image file formats", ACM Press, New York, *Addison Wesley*, (2004).
[12] Foley, James and Others," Computer Graphics Principles and Practice", *Addison-Wesley*, Reading, MA, 1996.
[13] CompuServe, Inc, "Graphics Interchange Formats GIF Specification", CompuServe, Columbus, OH, (1989).
[14]Ziv, J. and Lempel, A., "A Universal algorithm for sequential data compression", *IEEE Transactions on IT*, Vol. 23, No. 3, May (1977)
[15] Ziv, J. and Lempel, A.," Compression of individual Sequences via variable rate coding", *IEEE Transactions* on IT, Vol. 24, No. 5, September 1978, PP 530-536.



Hamza Abbass Ali is an associate professor at the Faculty of Science and Information Technology at Isra Private University (IPU), Jordan. He got his B.Sc.in 1968 from Basrah University, M.Sc. and Ph.D. in 1973 and 1977 respectively, from The University of London, UK. Before joining IPU, he worked as associate professor at Zarqa Private University (Jordan), visiting professor at University of Aizu (Japan) and associate professor at Basrah University (Iraq). His research interests include Cryptography, Information and Computer Network Security, Artificial Intelligence, Neural Networks and character recognition.

Bashar M. Ne'ma is an assistant professor at the Faculty of Science and Information Technology at Isra Private University (IPU), Jordan. He got his B.Sc.in 1995 from Al-Manssor University College, Iraq and M.Sc. and Ph.D. in 1999 and 2004 respectively, from University of Technology, Baghdad, Iraq, in Computer science. Before joining IPU, he was working as an Assistant Prof. in the University of Technology, Baghdad, Iraq and was a member of the Scientific Society for Computer, Iraq. His main work and his research interested in the basic theory of Speech processing, Cryptography and Information Security, Image Processing, Motion Detection.