Image files formats and Measuring Image Quality
A comparative study and analysis of commonly used image file formats

By Al-kenany S.H.

Received on: 25 / 9 /2004
Accepted on: 30/ 12/2004

Abstract

The core objectives of this research are to identify, understand and describe range of workable–based methods for quantitatively measuring the quality of an image represented in various file formats. Moreover, it suggests methods for measuring an image quality in varied graphic file formats. The research describes current popular and available graphic file formats, and the compression methods are utilized as well as color spaces, etc. The type of image display in use also affects the visual appearance of images. Furthermore, the lower depth output and use of colors fail affects the magnification features.

الخلاصة

تتلخص الاهداف الجوهرية لهذا البحث في تعريف وفهم ووصف العديد من طرق التصنيع المعتمدة على نوعية صورة ما ممثلة بعدة صيغ للملف . ويقترح البحث طرق قياس نوعية الصورة بصيغ عديدة لملفات . يصف البحث ملفات الرسم المتوفرة و الشائعة، وطرق الضغط المستخدمة ، وفضاءات اللون ، ونوعية الصورة المستخدمة وكيف تؤثر في اظهار الصور المرئية. إن اقل عمق للاخراج قد يؤثر في تكبير الخصائص، وكذلك أستخدام سقوط الالوان.

1- Introduction

With the proliferation of new computer systems and applications to run on them, a multitude of file formats has also been introduced by manufactures; each of these has been designed for either a specific software application or for a certain type of use. Today's computer-intensive society, most of us deal with computers on a regular basis, and computer graphics are being created and manipulated by users at every level of experience.[1][2].

Any person who knows generally about bitmaps without knowing anything specific about the formats compression

Lecturer assistant / University of Technology

techniques and overall structure of the graphic images will derive benefit from this paper. The paper will be used in measuring the quality of images and how this can be affected by the right or wrong choice of a file format. Moreover, it will be used to know the principles of image storage and its related areas including compression and decompression, color spaces and conversion between color systems, image displaying conversion between file formats and some advanced techniques used to enhance compression ration, and how to allow such features as real time full motion video.

2- Description of current Image Quality Measures

Many factors which are beyond our control affect the way we view image representations, some of the more distinct ones include equipment using a low–quality monitor with a poor graphics card which can only display say 16 colors at 320 × 200 pixels and will place an unfair disadvantage on all the format involved in the test.

Human vision:- Many people require man–made aids to help their vision nowadays.

Environmental conditions:- lighting is the main factor in this group.

Viewer bias:- For one reason or another an individual may have preconceived ideas about which format they believe will perform better, this already places bias towards the format before they have even been seen. Ensuring objectivity would be important and difficult in such tests.[3]

3-Measuring Image Quality

In the same way in any mathematical problem, measuring image quality involves a combination of many factors which affect the output quality of an image.

3.1 Factors Affecting Image Quality

With image format there are features and restriction: [1][4][5]

- Pixel resolution
- Color depth
- Compression and Decompression
- Color space
- Multiple Image storage

There are higher Level Factors

- Speed: While some formats provide great speed at representing and other storing images require more time for processing.
- File size: An uncompressed file seems to be larger especially if it is in 24-bit color.

- System resources.
- Application

4– File formats and compression methods

4. 1 format types

4. 1. 1 Vector: - Representation is performed by retrieving vector information for the image as objects. All the simple vectors objects can be stored as a set of mathematical statements with attached colors and other information. Vector format is well suited to storing mechanical and other technical drawing. This type of format would be unsuitable for storing real world scenes. This would require much processing and artificial intelligence to identify possible object outlines as the image is described as a set of formulas. Text display can not be guaranteed to be identical on different machines. Different machines may interpret the commands in slightly different ways as with a non–procedural programming language. Usually vectors are drawn in the same original order they were created in, and so complex drawings may give rise to a building block effect as the drawing is created before your eyes.

4. 1. 2 Bitmap:- This is the most widely used type that represents an image at a bit-by-bit level. Mapping of an image is made and transformed into pixel values which can then be assigned a colored value. Bitmap images are ideal for storing real-world photographic or other complex images as there is no requirement to classify objects, and as each pixel can be easily converted to an array value they are ideally suited to programming methods. These reasons have made Bitmap images popular with many differing applications. First thing to note about bitmaps is the large increase in dalg required to represent a scene for an accurate depiction of say a photograph resolution of at least 640×480 (307,200) pixels that would be required. At least 256 colors (8-bit) would be necessary for realism so the minimum storage required for the raw data is 307,200 bytes without even considering the color palette entries. Currently, resolutions up to 1600×1200 are possible at up to 128-bit true color requiring 30,720,000 bytes to store the image uncompressed. Storage space and image display memory are of much higher concern with bitmap data than it was with vector data. This remains a significant disadvantage of bitmaps; some of the industries where bitmaps are used, require manipulations of such data to identify object out lives and perform other operations to extract certain in formation from an image.

4. 1. 3 Metafile:- Beyond the vector and bitmap format now available this type fills a niche which bridges the gap between them. Metafiles have capability of handing both bit map and object vector data with in the same file.

The advantage of doing this is that some machine may only be able to display bitmap in formation while other can only cope with vector data. Both can be created for with metafile dramatically increasing the accessibility of the format type. In addition, this greater platform reduces the required file storage that may be in effect (only one file is needed for bitmap and vector representations of the same scene). The cost for this flexibility is transferred into construction and specification of formats. Meta file formats are generally more complex than bitmap or vector ones as they have to account for many more possibilities in content encoding or decoding other storage dilemmas. The knock on effect could be the decrease in the speed of encoding or decoding a complex file which utilizes the full range of facilities offered by the format. Finally, the high redundancy in ASCII files makes for high compression ratios. A widely known example of metafile format is that used by Microsoft Windows.

4. 1. 4 Scene description:- Formats of this type are almost indistinguishable from vector types and the only difference is that scene description files describe how to reconstruct the image as a whole. Individual objects are the same as vector and often it will be difficult to classify the two as separable groups.

4.1.5 Animation:- when a static image is no longer enough for displaying graphical information the next step is animation. Such formats store multiple static images in the same file which by displaying in quick succession give the appearance of movement. MPEG is a commonly used example.

4.2 Bitmap compression algorithms:

1. Symmetric and Asymmetric:

Symmetric codes are those which perform the reverse operation of a compression for decompression.

2. Non adaptive, Semi adaptive and adaptive encoding.

If the compression is to be performed on a specific type of data, we can decide that before use.

Semi-adaptive: The neat progression from a predefined dictionary is to have some idea of the contents which are to be compressed.

Adaptive: The most efficient of the three adapts. The dictionary entries are necessary whilst they move through the data [2]

3. Lossless v. Lossy

4. Pixel packing

5. Run length encoding

6. Lempel-Zivwelch (lZW)

7. Hoffman coding

8. Arithmetic coding [3][5]

4.3 Advanced Image Format

In this study, we look at a number of well-known image formats, and compare their features.

4.3.1 File Format Features

There are many different image file formats, and they all have different features and capabilities. In this section, we describe a superset of the features available in the most common image formats, divide the features into 4 types, inherent image properties, ancillary image properties, and compression and file organization.

4.3.1.1 Inherent Properties

Inherent image properties are those parameters which are essential to the interpretation of the image data. There are 3 basic properties you certainly need to know in order to interpret the raster data of a simple image. The first is the **Size** (width and height in pixels), the second is the **Color Model** (i.e. RGB, Gray etc), and the third is the number of **Bits per Color**. If you do not know these parameters, it is impossible to interpret the raster data (except by guessing the values). Some image formats support **Transparency**. There are two ways of implementing transparency, which are both in common use. The first is to define a transparent color, which allows you to make some pixels completely transparent while all other pixels are completely opaque. The second method is to add an extra color channel (often called an alpha channel), which allows each pixel to have a variable level of transparency. In order to correctly display an image, you need to know which type of transparency it uses, and which color/channel is transparent. Finally, some image formats allow an image to have **Extra Channels**. These can be used for various purposes for transparency information, or a mask which defines some selected area of an image, or special colors.

4.3.1.2 Ancillary Properties

Ancillary properties are not essential to basic use of the image, but contain additional general information about the image. Some formats have a fixed set of ancillary properties; other formats have flexible, extensible mechanisms for storing extra properties. The **Resolution** of an image, in conjunction with the pixel width and height of the image, indicates the intended or preferred physical size of the image. It is possible for an image to have different resolutions in the x and y dimensions, although this is not typically the case.

The **Orientation** defines how the image should be rotated for normal viewing. **Color Management** information consists of a (possibly large) set of parameters defining the characteristics of the source of the image (camera, scanner etc).

There are many types of additional information which are stored in some formats, including **Title**, **Description**, **Owner**, **Copyright**, **Date** and many others.

4.3.1.3 Compression

Most file formats support some form of data compression to reduce the size of the image data. A file format typically supports a limited set of one or more algorithms, selected to give good performance over the range of different types of image data it supports. Almost all compression algorithms are lossless. However, the JPEG algorithm is lossy (it throws away some of the image information during compression). The image you get back is not exactly the same as the image you started with. There is some loss of quality, but the advantage is that the compression achieved can be much higher.[5]

4.3.1.4 File Organization

Image files typically contain some form of header (containing information about the image), an index lookup table (if required) and the image data itself. In many cases, it makes little practical difference how these elements are arranged in the file. Some file formats provide the following features.

4.3.1.4.1 Random access.

Many simple formats store the image data in one large block. Typically, if the image is compressed, this means that you must decode the entire image even if you only wish to view a small part of it. This may not be a problem, unless the image is very large. In that case, it can be slow, and might use a lot of memory. Some formats allow the image to be split into a number of strips or tiles, each compressed independently, so that it is possible to extract an area of the image by just decoding the strips which are required.

4.3.1.4.2 Progressive Display

Most simple formats arrange the data as a sequence of scan lines, in order. If a large image is being transmitted over a slow network (e.g. an image on a web page), the user might see the image appear very slowly, line by line. From the users point of view, the initial image appears much faster, and then gradually improves in quality as the remaining scan lines arrive. This gives an impression of greater speed. In addition, the image becomes recognizable much earlier, so the user might be able to take action before the full image has arrived. There is a down side to progressive display it doesn't generally compress quite as well, so although the image starts to appear quicker, it might actually take slightly longer to totally complete.

4.3.1.4.3 Error Checking

Some formats have error checking built in to the format. This type of checking is useful in systems which have low error rates. Other formats feature error recovery, so that even if large parts of the file are lost it is still possible to make some kind of sense of the surviving parts.

A different view is that an error detection scheme which is closely linked to the underlying file format is more intelligent and robust. It can take account of the importance of different errors and can handle them appropriately. In the end, it depends on your application.

4.3.1.5 Abbreviations

Note the distinction between **Bpc (**Bits per color) and **Bpp (**Bits per pixel). For a non-indexed image, the **Bpp** is equal to **Bpc** multiplied by the number of colors (e.g., an RGB image with 8 bits per color would have 24 bits per pixel). For an indexed image, **Bpp** relates to the number of entries in the index table.[1]

4.4 BMP Format

The BMP image format is a native Windows image format. It is a very simple format which only supports RGB images (although there are ways to represent gray and bi-level images reasonably efficiently). The format can represent indexed images at up to 8 bits per color. There is also good support for small palette sizes. Compression is either raw, or a simple run length compression scheme.

4.4.1 Details

- **Inherent Properties**

Only RGB images are supported, and there is no support for transparency or extra channels.

Color Space	Bpc	Bpp	Idx	Comp	Notes
RGB	8	1	Yes	None	This creates a bi-level image with only 2 possible colors. The image data is stored efficiently using 1 bit per pixel.
RGB	8	4	Yes	None, RLE	This image is limited to just 16 colors. The image data is stored efficiently using 4 bits per pixel.
RGB	8	8	Yes	None, RLE	256 color indexed image.
RGB	5	16	No	None	Raw 5 bit RGB image.
RGB	8	24	No	None	Raw 8 bit RGB image.
RGB	8	32	No	None	This format contains a raw RGB image, but each pixel is stored as 4 bytes. This mode is not very useful in it own right, but it appears to be an attempt to add in forwards compatibility

Gray images can be supported by using a gray palette in RGB 8 Bpp mode. In that case, palette index *N* produces the RGB values for gray level *N* (i.e. R=*N*, B=*N*, G =*N*). Index values and gray values can be used interchangeably. A grey palette looks like:

Index	R	G	B
0	0	0	0
1	1	1	1
.	.	.	.
255	255	255	255

Similarly, bi-level images are supported using RGB 1 Bpp mode. This time the palette contains just 2 entries, one black one white.

- **Ancillary Properties**

Bitmap format can optionally store the resolution and orientation of the image data. Orientation is limited to top down or bottom up.

- **Compression**

RLE compression uses a simple proprietary run length coding scheme.

- **File Organization**

The format consists of a small header, a color lookup table (if required), and the image data. The header has a fixed structure with predefined fields.

4.4.2 Version 5 Additional Features

At the time of writing, the most recent version of the BMP format was V5, supported by Windows NT 5.0 and Windows 98 (or later). This adds a number of new capabilities. 16 bpp and 32 bpp modes allow different numbers of bits per color. For instance, in 16 bpp mode you could choose to use 6 bits each for red and green, and use the remaining 4 bits for blue. 16 bpp and 32 bpp modes also allow an optional alpha channel for transparency information. Once again, it is possible to allocate different numbers of bits to each of the RGB and Alpha channels.

- JPEG compression is supported.
- The file can contain color profile information.

Strengths	Limitations
Well supported in the Windows environment. Simple format, with few compatibility problems. Good for very small images because the file header size is very small. Efficient support for images with limited colors.	Only RGB is supported. No support for transparency. The simple compression scheme performs poorly for larger image sizes. Little support for ancillary information.

In summary, the small overhead makes this a good choice for small computer images, but the lack of features makes it a poor choice for general imaging applications.

4.5 PNG Format

PNG format is primarily associated with online imaging applications, although it is also a viable alternative for BMP format. It supports RGB and Grayscale images, using ZIP compression. The format is designed for portability, and supports features such as transparency and progressive display, which are required for web page images. It can also include a wide range of additional information about the image, if required.

4.5.1 Details

- **Inherent Properties**

Bi-level, Gray and RGB images are supported. Transparency is supported as described below.

Color Space	Bpc	Bpp	Idx	Comp	Notes
Gray	1	1	No	Flate	Bi-level image.
Gray	2, 4, 8, 16	2, 4, 8, 16	No	Flate	Grayscale image.
RGB	8, 16	24, 48	No	Flate	Full color RGB image.
RGB	8	1, 2, 4, 8	Yes	Flate	Indexed RGB image. The palette holds up to 256 entries.

PNG supports transparency in several ways:

• For non-indexed images with 8 or 16 bits per color, a transparency channel can be used. This means that each pixel can be fully or partially transparent.

• For indexed images, each color in the palette can be given a transparency value. Each color can be fully or partially transparent.

• For non-indexed images of any type, a single transparent color can be defined. The defined color is fully transparent; all other colors are fully opaque.

4.5.1.2 Compression

Flate compression is a general purpose adaptive compression algorithm used by the ZIP file format. PNG can also apply simple filters prior to compression, which can help to improve compression for some types of image data. Note that PNG images are always compressed; there is no option to leave the image uncompressed.

4.5.1.3 File Organization

The format consists of a small header, followed by image parameters, and sections of image data. Image parameters are stored in named chunks. This approach allows for optional parameters which can be left out if not required, and also allows for new chunks to be defined in a compatible way. PNG format can store a lot of optional information about the image, such as the resolution and orientation of the image data, the title, copyright and many other items. The image data can be broken into more than one chunk, but this does not normally allow random access into the image, because the chunks boundaries do not correspond to separately compressed strips of image data. The data is split only to help reduce memory requirements.

Strengths	Limitations
Well supported across various platforms. Simple format, with few compatibility problems. Supports transparency and progressive display. Supports ancillary information.	Only RGB and Gray color spaces supported. No support for efficient (JPEG) compression of photographic images.

4.6 TIFF Format

TIFF format specifies a set of baseline features which all readers are supposed to support. It also specifies some additional extensions which are optional. This description deals with baseline TIFF, with an extra section to describe the extensions. TIFF format is widely used in the graphic arts industry. It supports a variety of different image types and compression methods. It can also include a wide range of additional information about the image, if required. The format also provides control over how the image data is ordered.

4.6.1 Details

4.6.1.1 Inherent Properties

Bi-level, Gray and RGB images are supported. Transparency is supported, but by default the "associated alpha" model is used, (i.e.) the color planes are pre-multiplied by the transparency value. Extra color planes are also supported, but the baseline format doesn't provide any way to define what the extra planes mean.

Color Space	Bpc	Bpp	Idx	Comp	Notes
Gray	1	1	No	None, G3, Packbits	Bi-level image.
Gray	4, 8	4, 8	No	None, Packbits	Grayscale image.
RGB	8	24	No	None, Packbits	Full color RGB image.
RGB	164	4, 8	Yes	None, Packbits	Indexed RGB image. The palette holds either 16 or 256 entries.

4.6.1.2 Ancillary Properties

TIFF format can optionally store the resolution and orientation of the image data. It can also store the title, copyright many other items in its standard defined tags. In addition, it is possible to define private tags to store any other proprietary information which TIFF doesn't normally support.

4.6.1.3 Compression

Pack-bits compression is a simple run-length coding scheme. But it is specifically designed for bi-level data.

4.6.1.4 File Organization

The format consists of a small header, followed by image parameters, and sections of image data. Image parameters are stored using a dictionary of tag/values. This approach allows for optional parameters which can be left out of the dictionary if not required, and also allows for new tags to be defined in a compatible way. Various types of data can be stored, including integers, strings or arrays.

The image data can be stored as one or more strips. This provides a degree of random access into the image, because each strip can be decoded independently.

If there are multiple strips, they can be stored in any order.

Strength	Limitations
Supports many different types of image. Widely used in high-end graphic arts environments. RaRandom access to imimage data. SuSupports traTransparency. SuSupports ancillary infinformation.	Format is complex with various optional features, so compatibility problems can occur. Headers are relatively large, therefore not efficient for icons or very small images. Patent issues with LZW.

4.7 JPEG/JFIF Format

JPEG format specifies a set of baseline features which all readers are supposed to support. It also specifies some additional extensions which are optional. JPEG format is commonly used to store images in a variety of different application areas. To be strictly accurate, the file format is called JFIF, while JPEG is actually the name of the compression algorithm which the file uses. However, the files are almost universally referred to as JPEG files, and that is what we will call them here.

4.7.1 Details

4.7.1.1 Inherent Properties

Gray, RGB and CMYK data is supported, with a depth of 8 bits per color.

Color Space	Bpc	Bpp	Idx	Comp
Gray	8	8	No	DCT
RGB	8	24	No	DCT
CMYK	8	32	No	DCT

4.7.1.2 Ancillary Properties

JPEG supports few ancillary properties. It can indicate the image resolution.

4.7.1.3 Compression

Baseline JPEG format uses the JPEG DCT based compression algorithm. This is a lossy algorithm (that is, if you compress an image and then decompress it, you will not get back to *exactly* what you started with). However, it can achieve very high compression ratios with very little noticeable loss of quality. It is possible to vary this trade off. You can choose to retain a very high level of quality with a moderate amount of compression, or you can choose a lower level of quality but much higher compression. JPEG is designed to work with photographic images of natural scenes. The algorithm works by carefully discarding information from the image where it will not be readily noticeable. The reconstructed image is different from the original, but the differences don't stand out to the eye. If you turn up the compression (lowering the quality), then some visible differences (often termed artifacts) appear. Flat tints in the image can show a block pattern, which sharp high contrast edges can display ghosting or "ringing" effects.

4.7.1.4 File Organization

A JPEG file contains a sequence of parameter blocks and coded data blocks. The DCT compression algorithm divides the image into small tiles (typically 8 pixels square), and each tile is compressed independently. This allows random access into the image, and even allows certain operations to be performed without decoding the data. [4]

Strengths	Limitations
Very good compression ratio. Ability to choose balance between quality and compression. Widely supported. Random access to image data.	Compression is lossy, Works well with certain types of image. No support for index. No transparency support. Little ancillary information. Algorithm is relatively complex and computationally expensive.

5- Discussion

If we measure the spectral distribution of the reflected light under different conditions, it clearly varies a lot, yet the human visual system seems to report a constant color, the surface color. The problem is that for computer vision, we need to apply color constancy to be able to extract the surface color information which is the one we are interested in. Unfortunately, there is no complete algorithm for this (and no complete physiological theory either). See Fig (xx) in appendix A. We have 2 different images (A and B) in appendix A. The results are explained in the table 1 in appendix A. In table 2, we try to summary some graphics and file formats of the some properties.[6] In appendix B, we show some JPEG images with different ratios to explain the above-mentioned types of compression.

6-Conclusions

For digital image storage formats, digital image collections intended for long term storage and presentation should store three or four images or each original item: an archival image, derivatives for viewing and a thumbnail for browsing. The master or archival image should capture as much information as possible to preserve the investment in the capture process. Masters should use color rather than grayscale when color is an integral part of the information conveyed by the original object, and any compression applied to the file should be lossless. Viewing files can be created at any time from the archival image and should be created to provide reasonable access by standard viewers. It is recommended that at least two viewing files be created, a preview or thumbnail file for the fastest access during initial search and retrieval process and a service or reference image for more detailed viewing.

The type of image dislay in use also affects the visual appearauce of images. When using asset-up which can cope with more than 256 colors on screen at once no proplems arise. However, on lower color depth displays (255 colours or less) true–color images are not display correctly. This is because the images would require quantization to reduce the 24-bits of color information to 8-bits as suitable for display. Furthermore, such lower depth output also affects the magnification features, so the colors fail when used only when the image is normalized in both axes, and the color of 8-bit images return to its correct state. Color reflection is obtained by taking the integral over all the wavelengths. So, major limitation of the use of color in practice is in "surface color" and "illuminant color".

7-References

[1] Jackson Richard, MacDonald Lindsay, and Freeman Ken, "Computer Generated Color: A Practical Guide to Presentation Display", Glasgow, Scotland, 2000.

[2] Murry James D., and William Van Ryper, "Encyclopedia of Graphics File Format", Sebsat file format opol, CAO, Reilly and Associates, 1994, pp. 460.

[3] M. Gross, "Image Quality", John Wiley and Sons, 1999.

[4] Morello publishing, "Image File Formats: A Comparative survey", 2002.

[5] Nelson Mark, "The Data Compression Book ", New York: Mand T books, pp. 221-360, 1998.

[6] Sprigg, Graham, "Graphics File Format", New York: Mand T books, 2000.

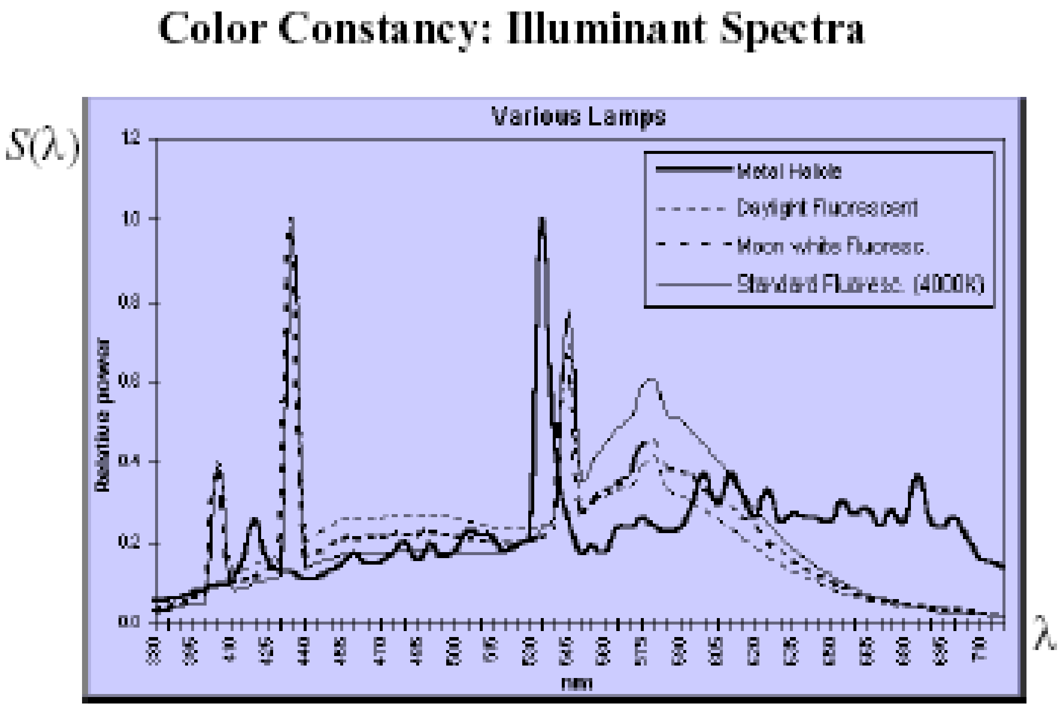
Appendix A

(a) (b)

Table 1

	Format	File size(bytes)	Compression (%)		
A	Bmp	19.320	Benchmark		
	Bmp(+RLE)	760	3.93		
	GIF	947	4.90		
	JEPG(max)	5.550	28.73		
	JEPG(low)	3.671	19.00		
	PCX	2.177	11.27	Load(secs)	Save(secs)
B	Bmp	4.285.232	Benchmark	.21	,21
	Bmp(+RLE)	N/A*	N/A	N/A	N/A
	GIF	768.934[1]	17.94	.14	.16
	JEPG(max)	348.408	8.13	.32	.27
	JEPG(low)	74.811	1.75	.25	.20
	PCX	3.915.808	91.38	.28	.26

*24-bit RLE compression is not generally supported in BMP 1Quantised to 256-entry color palette



Fig(xx)

Appendix B

Original image

Compression 8:1

Compression 18:1

Compression 53:1

Table 2

	BMP	CGM	DXF	EPS	GIF	IMG	JPEG	PCX
Name	Windows bitmap	Computer graphics metafile	Drawing Interchange Format	Encapsulated PostScript	Graphics Interchange Format	GEM Bitmap Image	Joint Photographic Experts Group	Z-soft PC paintbrush format
Dos file extension	BMP or DIB	CGM	DXF	EPS	CGF	IMG	(JPEG+TIFF)	PCX
Format Type	Bitmap	Meta file for vector and bitmap graphics	Vector	Vector and bitmap	Bitmap	Bitmap	Bitmap image compression	Bitmap
Versions	Windows3.x, OS/2 presentation manger	Depend on how the standard is implemented	AutoCAD release 12	Is a subset of Adobe PostScript	879 and 899	None	Various implementatio ns of the standard exits	Multiple
Variations	Windows and OS/2 BMP files vary in file structure ;most programs able to read both	3 different encoding schemes ,character ,binary, and text	ASCII and binary encoding schemes	Multiple	None			
Compatible O.S.	Intended for Intel Based PCS	Theoretically hardware neutral	Mostly PC,but supported on Macintosh	Macintosh, IBM pc,Unix,next	Most computer system	GEM graphical user interface for Atari ST and IBM PC	JPEG is a compression method that can be implemented by s.w. and h.w on virtually any computer system	Pc
Software that can open or input	Virtually all windows programs that utilize bitmaps	Most vector editing ,CAD, and page layout programs	All cad soft., many vector editing programs; some desktop publishing	Most desktop publishing and vector editing programs, some bitmap programs	Virtually all bitmap programs; most desktop publishing; vector editing software	Some windows desktop publishing and image editing programs	Recent versions of bitmap editing software and conversion programs	Almost all PC-based graphic application s
Color capabilities	2,16,256,or 16 m9llion colors	24-bit RGB,variable no. of colors in a palette	Everything from black and white to 16 million colors	Extensive-24-bit RGB and HSR,32-bit CMYK;gray-scale,indexd palette color	Indexed color palette, up to 256 colors	2,16,256,or 16 million colors	2,16,256,or 16 million colors;32-bit color depth	1-,2-,4-,8-,24-bit color; no grayscale
Compression	RLE sometimes implemented for 4- and 8-bit images	Different encoding schemes determine file size	Encoding schemes	Binary encoding	LZW run-length encoding	Simple RLE compression		RLE