



UNIVERSITI PUTRA MALAYSIA

**A COMPARATIVE STUDY ON COMPRESSION OF DIFFERENT
IMAGE FILE FORMATS**

OOI POH SAN

FSKTM 1999 11

**A COMPARATIVE STUDY ON COMPRESSION OF DIFFERENT IMAGE
FILE FORMATS**

By

OOI POH SAN

**Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of
Master of Science in the Faculty of Computer Science and Information Technology
Universiti Putra Malaysia**

September 1999



ACKNOWLEDGEMENTS

First and foremost, I would like to express my utmost gratitude to my project supervisor, PN. LILI NURLIYANA ABDULLAH, for her invaluable guidance, constructive suggestions and encouragement throughout the duration of this project.

I am also extremely grateful to Dr. ALI MAMAT and PN. RABIAH ABD. KADIR for checking this project. A special thanks to all the staffs of the Faculty of Computer Science and Information Technology who have been so helpful and supportive. I really enjoy working with them. Not forgetting also, to all my lecturers, thanks for everything.

Lastly, I would like to thanks my family for their undying love, patience and support which had enable me to complete my project successfully.



TABLE OF CONTENTS

		Page
ACKNOWLEDGEMENTS.....		ii
LIST OF FIGURE.....		v
LIST OF TABLE.....		vi
LIST OF IMAGE.....		vii
ABSTRACT.....		viii
ABSTRAK.....		x
CHAPTER		
I	INTRODUCTION.....	1
	Background.....	1
	Project Aim and Objective.....	2
	File Format in this Project.....	3
	Bitmap (BMP).....	3
	Tagged Image File Format (TIFF).....	6
	Graphics Interchange Format (GIF).....	9
	Tools and Equipment.....	10
II	LITERATURE REVIEW.....	12
	Basic Concepts of File Format.....	13
	Lossy Compression.....	14
	Lossless Compression.....	14
	Development of Compression.....	15
	Coding.....	15
	Modeling.....	15
	Compression Benefits.....	16
	Image Data Structures.....	17
	Image Coding Technique.....	17
	Entropy Encoding.....	19
	Source Coding.....	28
	Hybrid Compression Techniques.....	30
	Basic Concepts Of Scanner.....	32
	Bit Depth of an Image.....	33
	File Format in Scanner.....	34
	Raster File Formats.....	34
	Colour or Graytone File Formats.....	36
III	METHODOLOGY.....	38



IV	RESULT AND ANALYSIS.....	44
	Result.....	44
	Analysis.....	51
	Size Analysis.....	51
	Quality Analysis.....	56
V	CONCLUSION AND DISCUSSION.....	57
	Conclusion.....	57
	Discussion.....	59
	Future Work Study.....	60
	BIBIOGRAPHY.....	62
	APPENDIX	
	A.....	65
	B.....	67
	C.....	80
	D.....	82
	VITA.....	84



LIST OF FIGURES

Figure	Page
1.1 Read TIFF Header Function.....	8
1.2 Display TIFF File Image Function.....	9
2.1 Major Step of Data Compression.....	16
3.1 Main Interface.....	40
3.2 Compare (display) Image Interface.....	41
3.3 Show Bar Chart Interface.....	42
3.4 Convert Image Interface.....	43

LIST OF TABLES

Table	Page
3.1 A Rough Classification of Coding/Compression Techniques in Multimedia System.....	18
4.1 Size of Raw Image for Experiment-1.....	52
4.2 Size of Raw Image for Experiment-2.....	52
4.3 Size of Raw Image for Experiment-3.....	53
4.4 Compression Ratio for Experiment-1.....	53
4.5 Compression Ratio for Experiment-2.....	54
4.6 Compression Ratio for Experiment-3.....	54
4.7: Percentage of Size Reduction for Experiment-1.....	55
4.8: Percentage of Size Reduction for Experiment-2.....	55
4.9: Percentage of Size Reduction for Experiment-3.....	55



LIST OF IMAGES

Image	Page
4.1 TIFF-16 Colour	44
4.2 BITMAP-16 Colour.....	45
4.3 TIFF COMPRESS-16 Colour	45
4.4 TIFF 16 Colour	46
4.5 BITMAP-16 Colour.....	46
4.6 TIFF COMPRESS -16 Colour.....	47
4.7 TIFF-256 Colour.....	47
4.8 BITMAPP-256 Colour.....	48
4.9 TIFF COMPRESS -256 Colour.....	48
4.10 TIFF-256 Colour.....	49
4.11 BITMAPP-256 Colour.....	49
4.12 TIFF COMPRESS -256 Colour.....	50
4.13 TIFF Format.....	50
4.14 GIF Format.....	51



Abstract of the thesis presented to the Senate of Universiti Putra Malaysia in partial fulfilment of the requirements for the degree of Master of Science.

**A COMPARATIVE STUDY ON COMPRESSION OF DIFFERENT IMAGE
FILE FORMAT**

By

OOI POH SAN

June 1999

Chairman: Ali bin Mamat, Ph.D.

Faculty: Computer Science and Information Technology

Advances in imaging technology and computer communications have provided users with a variety of new services that use images, including video conferencing, videophones, multimedia system and High Density television. To fully utilize such a high tech communication system, image compression techniques play an important role in transmission and storage of information.

In this project, an image compression format and algorithm has been analysed. A system has been created for user to convert or display image file format.

The ScanJet 11C Scanner was used to scan the image. The image was coded into BMP, TIFF files, and GIF decoded and displayed on the screen. Some of the

experiment was done twice with two different types of images to ensure that the results were accurate. The Algorithms used to compress and decompress the image ware Run Length Encoding (RLE) algorithm and Lampel-ziv and Welch (LZW) algorithm.

One system was developed for users to convert the image file format to enable them to view the size of each image and to display image.

From the study, it can be concluded that LZW algorithm is better than RLE algorithm in term of percentage compression. Beside that, the quality of image that can be produced by LZW algorithm and RLE algorithm is almost the same.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi sebahagian keperluan untuk ijazah Master Science.

**PEMBELAJARAN PERBANDINGAN DARIPADA PEMAMPATAN IMEJ FAIL
FORMAT BERLAINAN**

Oleh

OOI POH SAN

Jun 1999

Pengerusi : Ali bin Mamat, Ph.D.

Fakulti : Sains Komputer dan Teknologi Maklumat

Kemajuan dalam teknologi penggambaran dan perhubungan komputer telah memberikan beberapa jenis perkhidmatan yang baru kepada pengguna seperti persidangan video, videofon, sistem multimedia dan "High Density television". Untuk memanfaatkan sepenuhnya sistem komunikasi berteknologi tinggi ini, kaedah pemampatan imej memainkan peranan yang amat penting dalam penghantaran dan penyimpanan maklumat.

Dalam projek ini, perisian pemampatan imej format dan algorithm telah dianalisis. Satu sistem telah dibangunkan untuk pengguna menukar atau memaparkan imej format.

ScanJet 11C Scanner telah digunakan untuk mengimbas imej, imej yang telah diimbas itu dikod di dalam fail Bitmap, TIFF fail, dan GIF dinyahkod dan dipaparkan pada skrin. Sebahagian eksperimen dijalankan dua kali dengan menggunakan dua jenis image yang berlainan untuk mendapatkan keputusan yang tepat. Algoritma yang digunakan untuk pemampatan dan nyahmampatan imej adalah algoritma “Run Length Encoding” (RLE) dan algoritma “Lempel-ziv and Welch” (LZW).

Satu sistem telah dibangunkan untuk pengguna menukar imej fail format, untuk menunjukkan saiz setiap imej dan memaparkan imej.

Secara keseluruhan pembelajaran ini, kita boleh membuat kesimpulan bahawa LZW algoritma adalah lebih baik daripada RLE algoritma dari segi peratusan pemampatan. Selain itu, kualiti imej yang dihasilkan oleh LZW algoritma dan RLE algoritma adalah lebih kurang sama.

CHAPTER 1

INTRODUCTION

Background

Nowadays we must handle large amounts of multimedia data such as text, image, video, and audio. This is one of the fundamental problems that needs to be addressed in multimedia. For example, CD-quality digital audio data (44 kHz sample rate and 16 bits per sample) takes 1 MB of storage space, as a single 640x480-pixel colour raster picture with high-quality (24-bits per pixel) colour. Compression is essential if audio, images, and video information are to be used economically in multimedia applications.

Because of bandwidth constraints, the principal focus of all coding work for audio and video information is concentrated on compression techniques. Compressing information to fit into the limited bandwidth available on current computer buses, local- and wide-area networks (LAN-WAN), and cable-television schemes is technically challenging and expensive.

Teleconsultation system is one of the system that using a lot of image has facing image compression problem. This system is aim to evaluate the ability of telemammography to facilitate real-time consultation between on-site general

diagnostic radiologists and remotely-located expert mammographers. Currently, such consultations are tedious, time-consuming, and logistically complex.

Nowaday, internet is widely use in every country. Because of internet contain a lot of image including the graphics and animation, people have to take a lot of time to open the websites or to download a data.

Project Aim and Objective

In today, many questions are raised in regard to getting a very good quality output from scanned images. These questions include: What resolution or bit-depth should we use when scanning the image? What file format should you use? Should it be RGB? And many other questions.

To match the above requirements, the aim of this project is to analyse an image compression algorithm suitable for scanning image. In order to achieve the project's aim, three objectives have been outlined in this project:

- To study and understand various kinds of image file format using different algorithm.
- To analyse the different image file format using the same algorithm.
- To create a system for user to convert the image file format, to view the size of each image and display image.

This project is mainly focus on three types of file format comparison that is Bitmap file format, TIFF file format and GIF file format. Beside that, bitmap file format and TIFF file format will divided into 16-colour and 256-colour. The comparison also includes brightness, colour and compression performance of the image. The objective for this project is for allowing people to refer which type of file format that they can choose. They can know the compression performance and the quality of image from this two type of file format.

File Format in This Project

Bitmap (BMP)

The simplest form of bitmap 4-bit and 8-bit images formats is using Run Length Encoding (RLE) algorithm in which recurring pixels are stored as a single pixel and a count value. The compression ratio thus obtained is highly variable and is dependent upon the recurrence of pixels (eg if there is a background of a single colour then a very good compression ratio can be expected, but if the image is highly irregular then it will be much poorer.

- Compression of 4-bit-per-pixel is using run-length encoded format for a 16-color bitmap.
- Compression of 8-bit-per-pixel Bitmaps is using a run-length encoded format for a 256-color bitmap.

BMP Bit-Depth

- 1-bit Bitmap is monochrome and the color table contains only two entries. Each bit in the bitmap array represents a pixel. If the bit is clear, the pixel is displayed with the color of the first entry in the color table. If the bit is set, the pixel has the color of the second entry in the table.
- 4-bit Bitmap has a maximum of 16 colors (2^4). Each pixel in the bitmap is represented by a 4-bit index into the color table array. The first pixel contains the color in the second table entry, and the second pixel contains the color in the sixteenth table entry.
- 8-bit Bitmap has a maximum of 256 colors (2^8). Each pixel in the bitmap is represented by a 1-byte index into the color table.
- 24-bit Bitmap has a maximum of 16.7+ million colors (2^{24}). Each pixel is a 3-byte sequence in the bitmap array represents the relative intensities of red, green, and blue.

File Format Specification

All bit map files contain a common header file which is declared as `BITMAPFILEHEADER` in `windows.h` [7]. It is declared as follow:

```
typedef struct tagBITMAPFILEHEADER
```



```

{
    UNIT    bfType;           // Identifier, must be "BM
    DWORD   bfSize;          // Size of file in bytes

    UNIT    bfReserved1;     // Reserve, must be zero
    UNIT    bfReserved2;     // Reserve, must be zero
    DWORD   bfOffbits;       // Offset after header where image begins
} BITMAPFILE HEADER;

```

A program can test to ensure that a file is in BMP format by checking the bfType data element. It will contain the string BM for bit map files. The other important element is the bfOffBits field which contain the distance in bytes from the end of the header (byte 14) to the beginning of the image data bits. This makes it easier to reach the beginning of the bit map bits.

Following file header is the bit map header and, optionally, the color map. The bit map header structure is declared as follows:

```

typedef struct tagBITMAPINFOHEADER
{
    DWORD    biSize;          // Size of the header
    LONG     biWidth;         // Image width in pixels
    LONG     biHeight;        // Image height in pixels
    WORD     biPlanes;        // Number of image planes, must be 1
    WORD     biBitCount;      // Specifies the number of bits per pixel
    DWORD    biCompression;   // Compression type
    DWORD    biSizeImage;     // Size in bytes of image
    LONG     biPelsPerMeter;   // Horizontal resolution, in pixels / meter
    LONG     biYPelsPerMeter  // Vertical resolution, in pixel / meter
    DWORD    biClrUsed;       // Number of colors used
    DWORD    biClrImportant   // Number of important colors
} BITMAPINFOHEADER;

```

The `biCompression` member specifies the type compression used for a compressed bit map image. It can be set to one of the following values:

- | | |
|----------------------|--|
| <code>BI_RGB</code> | Specifies that the bit map is not compressed. |
| <code>BI_RLES</code> | Specifies a run-length encoded format for bitmaps with 8 bits per pixel. The compression format is a 2-byte format consisting of a count byte followed by a byte containing a color index. |
| <code>BI_RLE4</code> | Specifies a run-length encoded format for bitmaps with 4 bits per pixel. The compression format is a 2-byte format consisting of a count byte followed by two word length colour indexes. |

Tagged Image File Format (TIFF)

The Tagged Image File Format known as TIFF, was developed by Aldus and Microsoft Corp. TIFF compress file format is using LZW algorithm. The basic purpose for which TIFF was created assumes that applications software for scanning or painting creates a TIFF file, which can then be read and incorporated into a document or publication by an application such as a desktop publishing package.

TIFF files exist in both compressed and uncompressed formats. The compression format offers a high degree of compression. TIFF files come in monochrome, 16-color, 256-color, 16-color grayscale, 256-color grayscale and true color (24-bit) varieties.

TIFF Bit-Depth

- 1-bit TIFF is monochrome.
- 4-bit Bitmap has a maximum of 16 colors (2^4). Each pixel in the TIFF is represented by a 4-bit index into the color table array.
- 8-bit Bitmap has a maximum of 256 colors (2^8). Each pixel in the TIFF is represented by a 1-byte index into the color table.
- 24-bit Bitmap has a maximum of 16.7+ million colors (2^{24}). Each pixel is a 3-byte sequence in the bitmap array represents the relative intensities of red, green, and blue.

File Format Specification

Among all the images files format, TIFF format is the one that can contain the largest amount of information in its header. Its header is constructed by three unique data structure. This makes it more complicated to access the images. Hence, it has the longest source code in the viewing program compared to others. Figure 1.1 shows how to access the header of the file.

```
/*read in the header*/
numberType=fgetWord(fp);
if(numberType=='MM' || numberType=='II')
```

```

{
if ((TIFFversion=fgetWord(fp)==42)
{ TIFFOffset=fgetLong(fp):
  fseek(fp,imageStart+TIFFOffset,SEEK_SET):
  TIFFEntries=fgetWord(fp):

if (fread((char*)&e,1,sizeof(EPSFHEAD).fp)==sizeof(EPSFHEAD))
{

if (!memcmp(e.epsf,"\xe5\xd0\xd3\xc6".4))
{
if (e.tiff_start>0L)
{
imageStart=e.tiff_start:
fseek(fp.e.tiff_start,SEEK_SET):
}
else
/* if there is more than one strip. The offset point to */
/* the offset of the first strip - convoluted */
{
for (l=0L;l<TIFFstripent:++)
{
fseek(fp,imageStart+TIFFstripoff+(l*sizeof(long).SEEK_SET);
t=fgetLong(fp);
fseek(fp,t,SEEK_SET);
n=ReadTifLine(p,fp);
p=farPtr(p,(long)bytes);
}
}
return
}

```

Figure 1.1 : Read TIFF Header Function

The display image function for TIFF file is list below:

```

void ShowTifPicture(p)
char*P; {
unsigned int i,w,d;
if (width> screen Wide)
w=pixels2bytes(screen Wide);
else
w=bytes;
if (depth>screenDeep)

```

```

    d=screenDeep;
else
    d=depth;
for (I=0; I<d;++I)
    memopy(MK_FP(0xa000,I*80),farPtr(p,(long)I*(long)bytes),w);
ibuf[0]=screenWide-1;
ibuf[1]=((screenWide-1)>>8);
ibuf[2]=screenDeep-1;
ibuf[3]=((screenDeep-1)>>8);
for (I=0;I<screenDeep;++I)

{
    memset(ibuf+4+(I*(screenWide >>3)),0,screenWide>>3);
    memset(ibuf+4+(I*(screenWide >>3)),
        farPtr(p,(long)I*(long)bytes),w);}
putimage(0,0,ibuf,COPY_PUT);}

```

Figure 1.2 : Display TIFF File Image Function

Graphics Interchange Format (GIF)

Graphics Interchange Format (GIF) uses a form of file compression called Lempel-Ziv-Welch (LZW) to minimize file size and electronic transfer time. GIF image can contain 2, 4, 8, 16, 32, 64, 128 or 256 colours, which are stored in a colour palette or colour table within the image file. Each colour in the GIF colour table is described in terms of Red, Green and Blue (RGB) values, with each value having a range of 0 to 255. Therefore, the GIF format has access to over 16.8 million colours, yet a maximum of 256 colors out of a possible 16.8 million can be referenced within a single GIF image.

GIF Bit-Depth

* 1-bit Bitmap is monochrome and the color table.

- 4-bit Bitmap has a maximum of 16 colors (2^4). Each pixel in the GIF is represented by a 4-bit index into the color table array.
- 8-bit Bitmap has a maximum of 256 colors (2^8). Each pixel in the bitmap is represented by a 1-byte index into the color table.
- 24-bit Bitmap has a maximum of 16.7+ million colors (2^{24}). Each pixel is a 3-byte sequence in the bitmap array represents the relative intensities of red, green, and blue.

When the number of colours used is significantly less than the current bit depth, then a bit depth reduction can occur which will reduce the file size without having any effect on the quality of the image.

Tools and Equipment

Some of the hardware and software development tools required in this project are listed below:

- Pentium 166 MMX
- 2.1GB Hardisk
- 120MB Floppy Disk Drive
- 104 Win95 Keyboard
- 14" Digital Monitor
- Hewlett Packard, ScanJet 11C Scanner
- BJC-210Sp printer
- Visual Basic 4.0

CHAPTER II

LITERATURE REVIEW

K. B. Kau (1993/94) had used the DT 2867 frame grabber card together with camera to capture the image and the image is coded into PCX,IMG and TIFF files, decoded and display on the screen. Algorithms used to compress and decompress the images are Run Length Encoding (RLE) algorithm and Lempel-Zif-Welsh Algorithm (LZW).

As can be seen, the IMG file format will have the best compressed rate under RLE compression algorithm which has reduced the raw image size by 60%. Whereas the PCX format performance is quite acceptable in which the size was reduced by half. RLE algorithm is not a good compressor for TIFF format in which only 20% of the size is reduced.

According to K. B. Kau, 1993/1994, the file that saved have actually been precompressed by the RLE compression algorithm. The compression between the size of raw image data and the compressed image file is done by the formula given below:

Size Of Raw Image = Dimension Of Image x Bytes Per Pixel

K. B. Kau, 1993/1994 found that only the RLE algorithm has been successfully tested on the images and the LZW algorithm need some work and effort to find out the errors and the problems in the program.

Beside that, he also found that an image compression format that are suitable for use of video transmission over a high-speed network. K. B. Kau(1993/1994) mentioned that the size of IMG format is smaller than the other two format that is PCX format and TIFF format. Nevertheless the size of the TIFF format is the biggest within that format which are used in the project.

Basic Concepts Of File Format

Data compression is a process that data stored on a disk or transmitted through a communication channel will occupy less space or travel faster if it can be reduced in size first. Data compression is used mostly to handle data in formats that tend to consume large amount of storage or channel capacity: in particular images, sound and video. The objective is to effect a saving in the amount of storage required to hold, or the time required to transmit, a body of digital information.

Data (images, video, etc.) is compressed by special programs called data compression software. When compressed data is retrieved from a disk or received at the reception end of a channel it is decompressed by similar software. Data compression technique can be divided into two major families that is lossy and lossless.

Lossy Compression

Lossy compression is permissible to loose quality in transmission and storage while retaining the general contents of the data such as in picture, movie and sound transmission. The kind of permissible loss is mostly determine by physiological factors as the capabilities of the human eye, ear and brain. Lossy compression proves effective when applied to graphics image and digitized voice. By their very nature, these digitized representations of analog phenomena are not perfect to begin with, so the idea of output and input not matching exactly is a little more acceptable. Most lossy compression techniques can be adjusted to different quality levels, gaining higher accuracy in exchange for less effective compression. Until recently, lossy compression has been primarily implemented using dedicated hardware.

Lossless Compression

Lossless compression consists of those technique guaranteed to generate an exact duplicate of the input data stream after a compress or expand cycle. Lossless algorithm use the fact that certain data symbols on the input are more frequent than others. Variable length coder are designed to the input data with shorter coder being assigned to more frequent symbols or symbol sequences and longer coder to rare symbols. This is the type of compression used when storing database records, spreadsheets, or word processing files.