深圳大学考试答题纸

(以论文、报告等形式考核专用)  
二○ 一九 ～二○ 二〇 学年度第2学期

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程编号 |  | | 课序号 | |  | | 课程名称 | |  | | 主讲教师 |  | | 评分 |  |
| 学 号 |  | | | 姓名 | |  | | 专业年级 | |  | | | | | |
|  | | | | | | | | | | | | | | | |
| 教师评语： | | | | | | | | | | | | | | | |
| 题目： | | Bitmap Image Representation and Applications of Divergent Image Types | | | | | | | | | | |  | | |

**Bitmap Image Representation   
and Applications of Divergent Image Types**

**Abstract**

Computer images permeate through current cyberspace. We encounter images of different formats off and on. In common scenarios it’s needless to know the disparity between them, but when it comes to basic web design or image editing or any other professional tasks, being capable of telling the differences would be of paramount importance. In this paper, we commence from .bmp format, a relatively rare yet fundamental image type that backs the others. Then we introduce GIF, which is mostly used for short animation on the Internet. It supports lossless image compression, but has been substituted by PNG’s and thereafter has become somewhat obsolete in the mainstream. After that, we view JPEG’s quintessential lossy compression on a general perspective. Finally, we make comparisons over these 4 formats and outline their usages on top of their attributes.

1. **Introduction**

The 20th century witnessed numerous groundbreaking progresses in computer image field. From image storages to data compressions, multitudinous vanguards probed into these issues to strike a balance between huge file size and decent image quality. A bounty of image types thereby blossomed, including .bmp, .jpeg, .gif and .png, which have been vastly generalized and become ubiquitous nowadays. Captivated by the divergences and their characteristics, to enable visualizing future advance, we made excerptions on relevant studies and arranged information of these image types.

Bmp, in actual, is an acronym for bitmap, a fundamental image type that backs the other 3 formats. Reputed for its capability to preserve image perfectly, .bmp image is also notorious for its sometime unreasonable file size, which is costly and unacceptable in many situations in the early years. To address this, a team at the online services provider CompuServe led by American computer scientist Steve Wilhite developed the Graphics Interchange Format (GIF) in 1987, and a committee known as the Joint Photographic Experts Groups (JPEG) finalized a unified standard for lossy image compression. In 1995, with the advances of PC performance and demands for a replacement of GIF owing to patent issues and its natural defects that restricted image quality, .png was then put forward.

Amid plethora of disparate data storage and compression methods, these 4 image formats and their correlated techniques outplayed, playing roles in different situations. In this work, we will start from .bmp, or bitmap, the fundamental data structure that backs up raster image. Then we will have a glimpse of GIF and step into PNG, which substituted GIF to a notable degree. Finally, we introduce JPEG, one of the most influential image compression techniques. Similarities and disparity will be amplified.

1. **Bitmap And .bmp Image**

.bmp file is a basic image type backed solely by bitmap lacking of compression.

In [1], bitmaps are defined as a mesh of regular rectangular cells called pixels, each of which contains a color value.Bitmaps are fundamentally characterized by 2 parameters, the number of pixels it contains, and the color values in the rectangular cells.

A bounty of images stored in computers are backed by bitmaps. Applied in image representation, bitmaps are oriented horizontally and vertically. In tune with the common sense, pixels oriented horizontally (width of images) does not necessarily equals to the ones placed vertically (height of images).

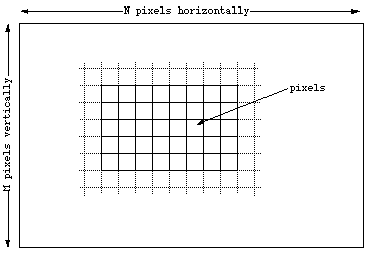


Figure 1. Basic bitmap structure. [1]

In the basic design, the number of bits used in each pixel determines color range of the image. For example, in an image with only black and white cells, we apply 1 bit to each cell. Value 0 is mapped to one color, and 1 is mapped to the other. On top of this, we could extend the bit size to 8, or one byte, then we derive the underlying skeleton of grayscale image.



Figure 2. A grayscale image.

One step further, we allocate 24 bits to each pixel, with each byte used to represent color depth of red, green and blue. The resulting pixel is capable of representing 256 \* 256 \* 256 different colors, trivializing distinctions between colors of similar RGB value. 16777216 different colors are sufficient to represent any images. Also, we call the number of bits used to represent colors Color Depth.



Figure 3. An RGB image.

In many scenarios, we also extend 24bit RGB to 32bit RGB. The additional 8bits constitute an 8 bit bitmap for transparency representation. We also refer it to as an alpha channel, a way to associate variable levels of transparency with an image. With the costs of extra bits, images are able to display various transparency, exhibiting details and ambiences on a better level. On this ground, we refer 32bit RGB to RGBA as well.



(*a*) (*b*)

Figure 4. Portrait with an oval alpha channel (*a*) against a white   
background and (*b*) against a black background. [5]

Nevertheless, as image size increases, thanks to the nature of bitmap, the file size grows substantially large, making it hard and even unacceptable for storage on PC in early years.

As a preliminary improvement, we change the bitmap structure to some degree, altering it to Palette-Based - Pixels in the bitmap do not represent RGB 3-tuple any longer. We list the color occurred in a specific image in a color table, or palette, and replace the value in pixels with the index, which ought to be able to be represented with less bits, to find the corresponding color in the table.

On the other hand, palette-based images do not own a full alpha channel in most cases. By far the most common approach to implement this indirectly is to specify that a single palette entry represents complete transparency. Consequently, some of the pixels that refer to the entry will be replaced by the image in the background. However, alpha channel is more widely applied in 32bit RGB, or RGBA images.

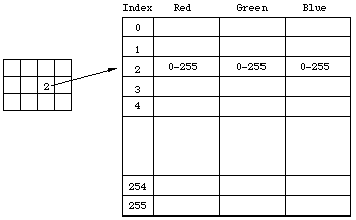


Figure 5. An 8 bit indexed color table. [1]

As is displayed in figure 4, the number of bits used to index color in the table is dependent of the diversity of colors. The saving-bit efficacy would be phenomenal if only few colors occurred in a large image. Bitmaps thus differs in the index value of each pixels and the correlated palette. However, it’s still a problem to select certain method, and in many cases, the improvement is far from satisfactory.

The amount of disk storage required for bitmap image could be roughly calculated in the following formula:

, where refers to the dimension, B the color depth in bits.

As an example, we choose an image of size 1K \* 1K, with the color depth being 24 bits (RGB). Size of the image is 3 MB, which is huge in the early years.

1. **Lossless and Lossy Image Compression**

Storing .bmp images on early PCs, which usually had limited storage, has been problematic because of its unacceptable size, putting image compression in the limelight.

* 1. ***The Graphic Interchange Format (GIF)***

.gif image, which is pervasive on the Internet nowadays, is backed by bitmap as well. It’s known for its support for short animations.

GIF is palette-based – it uses value in each pixel to index color and allows each image in the animation to have its own palette. In some scenarios, palettes are provided by the website on which the .gif is displayed.

Images are compressed using Lempel-Ziv-Welch (LZW) lossless data compression. The number of bits used to index color in the color table is fixed to 8, meaning that it supports 256 different colors in each image, which could restrict image’s visual quality.

Sometime we use techniques called dithering to turn a palette consisting of wide range colors into a smaller color table, making approximations on in-between colors to meet the color-range requirement. However, this inevitably results in data loss, generating noises in images, and is discordant to the main purposes of GIF in the meantime. Worse still, GIF only support palette-based image compression, which is based on LZW algorithm, a technique that was patented in the last century.



Figure 6. A gif image after applying dithering.

Despite the color range constraint, GIF is capable of storing true color image as well, though nowadays we usually use .png or .jpeg instead [2]. To do this, we split the original image into numbers of blocks, each of which has its own (256-color) palette that is able to store all the colors needed in the block (thereby making dithering unnecessary). Finally we integrate them, forming a true color image. Nevertheless, similar colors is probable to be merged, leading to data loss by different margins.

1. (*b*)

Figure 7. (*a*) The .gif with 255 colors and (*b*) with 1859 colors after applying the technique.

The main idea behind implementation of animated GIF is that we apply Graphics Control Extension (GCE) to postpone the timings of images to be painted. Each .gif image has its own GCE. By assistance of Netscape Application Block, .gif image is able to loop for a specified times or continuously. [3]

On the other hand, GIF does not support partial transparency. Alpha channel isn’t applied in the techniques. It permits only binary transparency – pixels in the image can either be fully transparent or opaque.

In addition, GIF supports one-dimensional interlacing, which enables image preview while images are loading. During the GIF painting process, interlaced image will displays in ordered 8 rows. Pixels in each row are presented consecutively from left to right. The image data is divide into 4 parts, that is to say, it makes it possible for users to recognize the original image after one fourth of the data is downloaded.

* 1. ***Portable Network Graphics (PNG)***

Owing to the natural defects of GIF and its infamous patent issues, with the advent and popularization of more advanced displayers, PNG, as a bitmap-backed raster image file format that supports lossless image compression, was developed and has replaced GIF in a vast field.

PNG supports palette-based, grayscale and full-color non-palette-based RGB or RGBA images. It is designed originally for image transfers on the Internet. Although PNG updated support for animation after its debut, the majority of .png files remain single-image [4], which is also its hallmark comparing to GIF.

In regards to transparency, PNG allows a maximum total of 65536 (216) transparency levels, but usually 256 different levels are adequate for common usages. The space demanded in comparison to .bmp are smaller.

Similar to but has outplayed GIF, PNG supports 2 dimensional interlacing, making it possible for better progressive display. Recall the time period when staring at a .png image being loaded, the entire image was first vague, but then gradually became clear. 2 dimensional interlacing succeeds in making the image being load symmetrically. More exactly, unlike interlacing in GIF in which the image was load by rows (which corresponds to horizontal resolution), vertical resolution of .png images are guaranteed. In general, the time needed to make textual information in an image readable for GIF is 2 to 4 times longer to for PNG.

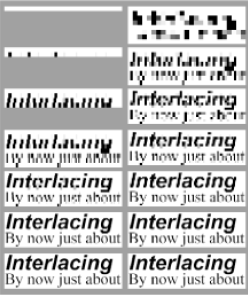


Figure 8. Comparison of GIF interlacing (left), normal PNG interlacing (right). [5]

Finally, we come to PNG’s lossless compression, the most determining charisma of PNG. PNG supports compression over true-color, grayscale and palette-based image. PNG compression is primarily comprised of 2 parts – Deflate and Compression Filters. Deflate, which capitalizes on LZ77 and Huffman Coding, supports different levels of compression, with the time required proportionate to it. Compressing image at the highest level, the time needed could be considerable. Compression filters in actual are a way of transforming image data that could render the resultant size smaller. There are in total 5 different filter types. Compression could be decently improved by opting the types wisely.

* 1. ***JPEG Image Compression***

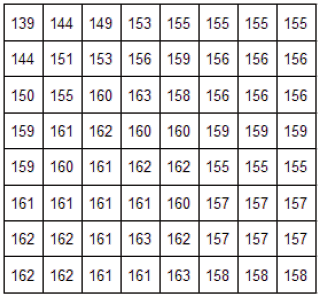
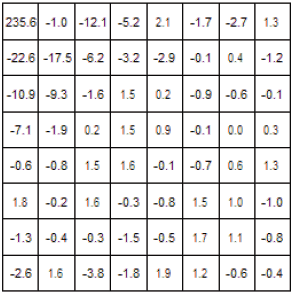
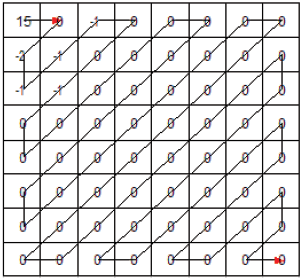
GIF and PNG are 2 well-known image types that could make lossless compressions over images. Nonetheless, size of .gif or .png files, sometime, is still a problem for PC storage and image data transfer on the Internet. Under lots of situations, image quality, though important, does not matter too much. Basing on this point, JPEG image compression, the storied lossy image compression, enables us to compress image to an extraordinary degree, preserving overall visual quality with details losses.

Comparing to GIF and PNG, JPEG format does not support short animations or interlacing, and does not store transparency information. Still, JPEG supports progressive display as PNG does. Also, JPEG compression could be applied to image that is backed by bitmap. Extraordinarily, JPEG’s technique supports prominent compression ratio that outstrips lossless compression, making itself universal and irreplaceable.

The main idea behind JPEG compression is tricky – it exploits the limitations of human eye, which did a poor job on detecting changes in the bright and dim color from pixel to pixel. JPEG compression parlays this into imperceptible details discard and thereby reducing the file size but does not affect the overall visual quality of the image.

Although JPEG derivatives abounds nowadays, baseline JPEG compression process consists of 4 phase in general – The Discrete Cosine Transform (DCT), Quantization, Reordering and Entropy Coding. Ahead of the compression, color space transformation may be undertaken, and the original matrix, to better utilize computing power, will be split into 8 \* 8 blocks.

DCT lies at the heart of the entire technique, transforming the original pixel matrix into the one that could be easily quantized and clustering the most important data around the top left of the matrix. Quantization is also where the real compression takes place, disposing of a hail of information (which results in zeros in the matrix as shown in figure 8 (*c*.)).

(*a.*) (*b.*) (*c.*)

Figure 9. (a) Original pixel matrix, (b) after DCT, (c) after Quantization and Reordering. [6]

By reordering, we obtain a row vector that contains a great number of zeros, which would then be replaced by a placeholder EOB, reducing bits needed and signifying that the vector is from an 8 \* 8 matrix. Finally, we apply Huffman Coding, an entropy coding technique that is applied by JPEG standard, to finalize the resulting image file. At a first glance, the difference between compressed image and the uncompressed one seems to be trivial. In some cases, quality of the resultant image lowers owing to faults in Huffman Table, which could be evaded by applying bi-residual codes [8] and further measures.

(*a*) (*b*)

Figure 10. (*a*) The original image and (b) the image after being compressed. [7]

1. **Merits and Drawbacks of BMP, GIF, PNG and JPEG.**

Commencing from .bmp, as is depicted, BMP images hold the raw image data unchanged, reserving all details and original information, but leave the file size colossal. GIF is capable of taking lossless compression over image, and outstandingly, it supports short animations, which make it a commonplace on the Internet. However, its performance is also constrained by the fact that it only supports 256 different colors. To this end, PNG lossless compression would be an option. Furthermore, at the cost of reasonable details losses, JPEG enables spectacular compression ratio.

On balance, .bmp image is most suitable for raw image storage on PC. Certain compression could be taken according to the requirements. For short animation and sharp-edged line art with a limited number of colors, such as logo, GIF may be the best choice. PNG can satisfy the needs for editing image and various transparency, for example, partial, on/off and no transparency. On top of lossless compression, PNG outshines others as well. If image details can be discarded to some degree and there is no demand for image transparency, JPEG lossy compression could also be taken into consideration. For both GIF, PNG and JPEG, all of them are of decent browser portability partially but significantly because of their sufficiently small file size.

1. **Conclusions**

In this paper, we investigate BMP’s fundamental solutions for representing and storing image and introduce 3 different image types, and the general techniques, finally shed some lights on their divergences and applications. In many situations, BMP image, though compressible, is too large to store, transfer conveniently. Thus GIF was proposed. But nowadays it’s used chiefly for short animations. PND outplays GIF when it comes to lossless compression. If details of images are reckoned insignificant, JPEG format would also be an option for storing images on PC and transferring them on the Internet.

**References**

1. Paul Bourke. (1993, Nov). *A Beginners Guide to Bitmaps*. [Online]. Available: <http://paulbourke.net/dataformats/bitmaps/>
2. Andreas Kleinert. (2012, March 16). *GIF 24 Bit (truecolor) extensions*. [Online]. Available: <https://web.archive.org/web/20120316215949/http://uk.aminet.net/docs/misc/GIF24.readme>.
3. Royal Frazier. (1999, April 18). *All About GIF89a*. [Online]. Available: <https://web.archive.org/web/19990418091037/http://www6.uniovi.es/gifanim/gifabout.htm>.
4. *PNG standard*, section 8.4, ISO/IEC 15948, IETF RFC 2083, 1997.
5. Greg Roelofs, “An Introduction to PNG,” in *PNG: The Definitive Guide* 2nd ed. Sebastopol, O’Reilly, 1999, ch. 1, sec. 1.2.1.
6. Robert G. Redinbo and Cung Nguyen. (2003, April 18). *JPEG Image Compression System*. [Online]. Available: <https://www.ece.ucdavis.edu/cerl/reliablejpeg/compression/>
7. Ian Snyder. (2009, December 14). *JPEG Compression*. [Online]. Available: <https://mse.redwoods.edu/darnold/math45/laproj/f09/isnyder/Presentation.pdf>
8. Cung Nguyen, “Fault Tolerant Huffman Coding for JPEG Image Coding System,” The ECE Dept., Univ. CA, Davis, Tech. Rep. CA 95616, Apr. 12, 2003.