Image Size Variation Influence on Corrupted and Non-viewable BMP Image

Tengku Norsuhaila T Azmi¹, Nurul Azma Abdullah², Nurul Hidayah Ab Rahman³, Isredza Rahmi A. Hamid⁴, Chuah Chai Wen⁵

Information Security Interest Group (ISIG), Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

Corresponding author: azma@uthm.edu.my

Abstract. Image is one of the evidence component seek in digital forensics. Joint Photographic Experts Group (JPEG) format is most popular used in the Internet because JPEG files are very lossy and easy to compress that can speed up Internet transmitting processes. However, corrupted JPEG images are hard to recover due to the complexities of determining corruption point. Nowadays Bitmap (BMP) images are preferred in image processing compared to another formats because BMP image contain all the image information in a simple format. Therefore, in order to investigate the corruption point in JPEG, the file is required to be converted into BMP format. Nevertheless, there are many things that can influence the corrupting of BMP image such as the changes of image size that make the file non-viewable. In this paper, the experiment indicates that the size of BMP file influences the changes in the image itself through three conditions, deleting, replacing and insertion. From the experiment, we learnt by correcting the file size, it can able to produce a viewable file though partially. Then, it can be investigated further to identify the corruption point.

1. Introduction

Digital forensics can be defined as the application of computer science and investigative policies for a illegal crime purpose involving the evaluation of digital evidence after proper search authority, chain of custody, validation with mathematics, use of validated tools, repeatability, reporting, and possible skilled presentation [1]. During Digital Forensic investigation, recovering image is one of the digital evidence acquiring techniques. Recover the files from damaged is an important technique to recover the digital evidence in these issue. Images are the most common file that have been used by forensic to recover from the target disk. Commonly, image files used in digital forensics as data evidence. JPEG is the most popular among the image formats used on the Internet. This is because JPEG files are very lossy and easy to compress that can advance in Internet transferring processes [2] [3]. However in some problems, these images are corrupted when stored in the hard disk. Corrupted JPEG images are hard to recover compared than BMP images due to the complexities of determining corruption point. Since JPEG files are very lossy, means that most of the information are discarded compared to the BMP image which are lossless file [4]. Therefore, the file is required to be converted into BMP format in order to investigate the corruption point.

BMP is used for display on the screen because it is in spatial domain. In image processing, BMP format images are preferred over other images, as they contain all the image information in a simple format. According to [5], BMP files can be edited and changed easily due to their high quality in image processing programs where it can be read and write, debugged and viewed without tools. It is used to display image file including the color of each pixel. A BMP image represents the colors in spatial domain where an image is corresponding with the values of pixels in an image [6]. In this paper, an experiment is done to view the structure of BMP file and understand behavior of BMP...
influenced by size. It can be able to summarize header structure, color table and pixel data in BMP file. From the experiment, it also can help researcher to understand the behavior of BMP files.

The rest of the paper is organized as follows. Section 2 describes related works of BMP, Section 3 discussed about the experiments have been done, Section 4 described about the result and discussion. Finally section 5 described conclusion and future works of this paper.

2. Related Work
In this paper, corrupted and non-viewable BMP image can view based on changing the BMP size. This section focuses on the history of BMP file, BMP pixel storage and the BMP structure.

2.1. Overview of BMP file
BMP is a standard image file format introduced by Microsoft acts as a medium between users and their windows operating system for display an image. Nowadays BMP file format is supported many file systems and operating systems but less popular because of its size which is larger compared to another file formats [7]. The BMP format also known as device independent bitmap (DIB) file format which is a simple raster graphics image used to store BMP digital images independently of the display device. It can store two dimensional images that is monochrome and colour in different colour depths (1, 4, 8, 16, 24, 32, or 64 bits per pixel) [8].

2.2. BMP pixel storage
In BMP file formats, image pixels are stored with a colour depth of 1, 4, 8, 16, 24, or 32 bits per pixel. Uncompressed BMP image stored much larger than compressed image file formats for the same image [9]. The main colour information is discussed below:

2.2.1. 1 bit (black and white). 1 Bit also called monochrome or black and white which is content the lowest possible data can be held for every pixel. There are only two possible colours. The pixels with a '0' refers to a black colour and pixels with '1' refers to a white colour.

2.2.2. 8 bit greys. Eight-bit colour which has 256 colours is stored one pixel value per byte. Every byte is an index into a table of up to 256 colors. In this case each pixel takes 1 byte (8 bits) of storage resulting in 256 different situations. It will show 0 is normally black and 255 white. For example of grey levels are shown in figure below:

![Figure 1. Grey level of linear scale](image)

Figure 1 shows that the numbers in between white and black pixel are the grey levels. For example in Figure 1, 0 is a black and 255 is a white while 127 would be a 50% grey level.

2.2.3. 24 bit RGB. In a BMP image, RGB colour (24-bit) pixel values are stored with bytes as BGR which is blue, green and red component. There is 8 bits assigned to each colour component. In each colour component, the value of 0 refers to no contribution while 255 refer to totally contribution. As every component has 256 different states there are a total of \([256]^3\) which is 16777216 possible colours of each pixel [10]. An uncompressed 24-bit BMP image is denoted by 24 bits or three bytes (24/8 bytes) of each pixel. Each byte contents 8 bits which match to the three colour planes which are red, green and blue [8]. A white pixel would be denoted in binary as 11111111 11111111 11111111, and a black pixel would be 00000000 00000000 00000000. 8 bits that allowed for 2^8 or 256 possibilities different colour.
2.2.4. 32 bit RGB. Usually 32 bit colour is the same as 24 bit colour but added with 8 bit BMP known as an alpha channel that represents the degree of transparency of a colour. This channel area controls the transparency or opacity of a colour. Alpha value is used to determine the resulting colour when an image is overlaid onto another image [11]. Example if the alpha value is transparent, the source colour is invisible.

2.3. BMP Structure

BMP file structure has been explained to understand how BMP image can be used to summarize the structure of BMP image. It has specific file structure which contains File Header, Info Header, Colour Table and Pixel Data is shown in figure below.

| Field: BMP File Format                                      | Name       | Size | Description |
|-------------------------------------------------------------|------------|------|-------------|
| FileHeader                                                  | 8 bytes    |      | Windows Structure: BITMAPFILEHEADER |
| Signature                                                   | 2 bytes    |      | BM          |
| FileSize                                                    | 4 bytes    |      | File size in bytes               |
| Reserved                                                    | 4 bytes    |      | Index (0)               |
| Offset                                                       | 2 bytes    |      | File offset to Raster Data       |
| InfoHeader                                                  | 50 bytes   |      | Windows Structure: BITMAPINFOHEADER |
| Name                                                        | 4 bytes    |      | Size of InfoHeader = 50          |
| Width                                                       | 4 bytes    |      | Bitmap Width                     |
| Height                                                      | 4 bytes    |      | Bitmap Height                     |
| Planes                                                      | 2 bytes    |      | Number of Planes = 1             |
| BitCount                                                    | 2 bytes    |      | Bits per pixel                   |
| Compression                                                 | 4 bytes    |      | Type of Compression              |
| 1 = BI_RGB no compression                                  |            |      | 1 = BI_RGB no compression        |
| 2 = BI_RGB 4:4:4 compression                               |            |      | 2 = BI_RGB 4:4:4 compression     |
| ImageSize                                                   | 4 bytes    |      | Compressed Bit Size of Image     |
| CompressedBitSize                                           |            |      | Not valid if size = 0            |
| PlanesX                                                     | 4 bytes    |      | Horizontal Pixels Per Meter      |
| PlanesY                                                     | 4 bytes    |      | Vertical Pixels Per Meter        |
| ColorTableSize                                              | 4 bytes    |      | Number of actually loaded colors |
| ColorSpace                                                  | 4 bytes    |      | Number of important colors       |
| 1 = all                                                    |            |      | 1 = all                          |
| ColorTable                                                  | 4 * NumColors bytes | NumColors = NumColors * 4 bytes | Table must be ordered by importance |
| Red had                                                    | 3 bytes    |      | Red intensity                    |
| Blue had                                                    | 3 bytes    |      | Blue intensity                   |
| Green had                                                   | 3 bytes    |      | Green intensity                  |
| Premultiplied                                              | 3 bytes    |      | Premultiplied                     |
| RasterData                                                  | InfoImageSize bytes |      | The pixel data                   |

Figure 2 shows the BMP file structure where each image file contains four sections. The File Header consist of general information about the file while the Info Header contains general information about the image in the file. Palette of colours used in the image are defined in the Colour Table. In the BMP file, the palette occurs after the BMP header. It is not used when using 16-bit or higher of BMP image. The contents of the dots in the image are identified in the Pixel Data by identifying, for every pixel that colour in the palette is used in the pixel [13]. A BMP file can be divided into two main blocks, the header and the data. The header that contains of 54 bytes can be divided into two sub blocks which are known as BMP Header and the BMP InfoHeader [8].
2.3.1. **BMP Header.** BMP header structure which is used to detect the file as a valid BMP image. The BMP Header description is described as in Table 1.

| Starting bytes | Size | Description |
|----------------|------|-------------|
| 0-1            | 2 bytes | Indicate the characters “BM” that classify the file type |
| 2-5            | 4 bytes | Size of file |
| 6-9            | 4 bytes | Reserved |
| 10-13          | 4 bytes | Starts of pixel data |

Table 1 shows the description of **BMP Header.** File header of BMP image has 14 bytes. In BMP file format, the first two bytes indicate the character ‘B’ followed by the character ‘M’. The following four bytes represent of the file size and next four bytes are reserved for application identification. The remaining four bytes represent the offset to start of pixel data where the BMP data can be found.

While the **BMP InfoHeader** represents of the next 40 bytes in a file which started at byte fourteen [8]. The information of **BMP InfoHeader** is described as in Table 2.

| Starting bytes | Size | Description |
|----------------|------|-------------|
| 14-17          | 4 bytes | BMP header size (40 bytes) |
| 18-21          | 4 bytes | BMP image width |
| 22-25          | 4 bytes | BMP image height |
| 26-27          | 2 bytes | Number of colour planes |
| 28-29          | 2 bytes | Number of bits per pixel which is the colour depth of the image (1, 4, 8, 16, 24, 32) |
| 30-33          | 4 bytes | BMP compression method being used (BI_RGB=0, BI_RLE8=1, BI_RLE4=2, or BI_BITFIELDS=3) |
| 34-37          | 4 bytes | Size of BMP image. Size of the raw BMP data (may be 0 if uncompressed) |
| 38-41          | 4 bytes | The horizontal resolution of the image in pixels per meter |
| 42-45          | 4 bytes | The vertical resolution of the image in pixels per meter |
| 46-49          | 4 bytes | The number of colours used |
| 50-53          | 4 bytes | The number of important colours used |
Table 2 shows the description of **BMP InfoHeader** in BMP image. The size of **BMP InfoHeader** is 40 bytes that shows specific information about the image.

Example of **BMP Header** and **BMP InfoHeader** are opened in Hex Editor shown as in Figure 3.

Figure 3 shows the **BMP Header** structure. The first two bytes represented by 0x42 and 0x4D in Hex Editor which denotes “BM” in ASCII. In second red box represented by 0x8A indicated the start of pixel data in a file. Third and fourth red box indicates the width and height in BMP file. In this experiment, the width and height are the same which is 0xEC (a square picture). 0xEC is 236 in decimal, which means this is a 236 x 236 BMP image. The calculation below shows how to determine the row size of BMP image.

\[
\text{RowSize} = 4 \times \left( \frac{\text{BPP} \times \text{Width}}{32} \right) \\
= 4 \times \left( \frac{24 \times 236}{32} \right) \\
= 708 \text{ bytes}
\]

Line (2) shows that the row size is bits per pixel (24) multiply by width (236 bytes) then divide by length of row (32) and multiplies by size of BMP row (4). Thus, the result is 708 bytes in line (3) denotes the row size.

**2.3.2. BMP Data.** The BMP Data block of a BMP file consist of the real image, saved as pixels. Example a snippet of the BMP Data block in a BMP image as shown in Figure 4.
3. Experimentation
In this section, some of the experiments have been done to summarize the structure of BMP file and behaviour of BMP image as described below.

3.1. Structure of BMP Image
In this paper, an image from the Internet is used to view the structure of the BMP image. Original image 5(a) is compared to a corrupted image of the same file in the 5(b).

![Original BMP and Corrupted BMP](image)

A corrupted and a non-corrupted file of the same BMP file are used to view the structure of BMP image. This picture has a resolution of 236 x 236 pixels and the file size is 167,226 bytes. For the example, the first row until 30th row has been changed from this experiment to show how BMP structure can be read.

To simulate corrupted BMP file, the original BMP is sliced into each rows which every row is contained of 708 bytes. In order to display corrupted file in Windows Photo Viewer, example of 30 rows are selected and changed with other data value that refer to Figure 7. As shown in Figure 5(b), 21594 bytes are changed in this test.

This experiment shows that the location of pixel of actual BMP image is different from where the pixel stored as shown in Figure 6(a) and 6(b). Example the first 30th rows are selected to show the structure location BMP pixel image.

![Structure location BMP pixel image](image)

Figure 6(a). Structure location BMP pixel image when deleted and replaced the first 30th row of file
Figure 6(b). Structure location BMP pixel image when deleted and replaced the last 30th row of file

Figure 6(a) and 6(b) show the testing of structure location BMP pixel image in the Hex Editor and normal image. From this experiment, when the first 30th rows of data has been deleted and replaced with another, the image becomes corrupted at the bottom. While the image becomes corrupted at the top by deleting and replacing the data in the last 30th row of file.

3.2. Behavior of BMP image

The same image in Figure 5 is used to test the behavior of BMP image by deleting, replacing and adding data with another BMP image refer to Figure 7. The experiment is tested and described in the table below.
Table 3. Experimentation of BMP size variation

| Original BMP image | Conditions                          | Size          |
|--------------------|------------------------------------|---------------|
| Size: 163 KB       |                                    |               |
| Dimensions: 236 x 236 | Delete the first row, 708 bytes data of original BMP image. | 162 KB (Less than the original image) |
|                    | Condition 2                        |               |
|                    | Replacing data, example the first 30th rows, 21240 bytes data of original image with another BMP data (refer to Figure 7). | 163 KB (Same with the original image) |
|                    | Condition 3                        |               |
|                    | Adding data, example the 30 rows from another BMP data (refer to Figure 7) into the original data image. | 184 KB (more than the original image) |

Based on Table 3, 236 x 236 square BMP image are used to test the behavior of BMP image. The experiment has been divided by three conditions: The data which is deleted from original file, replacing the original data with another data and adding another data into the original file. First condition is deleting the first 30th rows of original BMP image and the size of file become smaller than the original image. The second condition is replacing the first 30th rows of original data with another data, hence the size must be the same with the original image. Last and third condition, the size is more than the original image by adding 30 rows from another data.

4. Results and Discussion
This Chapter discusses about the structure of BMP file and result of behavior BMP image. The results have been described below:

4.1. Structure of BMP image
A BMP file consist of BMP structure, color table and pixel data to summarize BMP file structure. As mention on the Subsection 3.1, it can summarize that pixels are stored “upside down” when open using Hex Editor while in normal image, it was started from the lower left corner to right, and then going through row by row from the bottom to the top of the image. Figure 8(a) and 8(b) show how different viewed in Hex Editor and screen view.
Figure 8(a). BMP file open in Hex Editor

Figure 8(b). BMP file on screen

The order of the pixels in BMP file is started from the left to right, bottom to top which is first pixel is from lower left corner of the image. Based on reading data from memory, the actual image can be displayed upside down.

4.2. Behavior of BMP image
Changing the size can influence the corrupting of BMP image. The experiment done in Section 3.2 and the result is described on the table below:
### Table 4. Result of BMP size variation

| Original BMP image | Conditions | Result | Size       |
|--------------------|------------|--------|------------|
| Size: 163 KB       | **Condition 1** |        | 162 KB     |
| Dimensions: 236 x 236 | Delete the first row, 708 bytes data of original BMP image. | Image cannot be displayed and error message is displayed instead. | (Less than the original image) |
|                    | **Condition 2** |        | 163 KB     |
|                    |Replacing data, example the first 30th rows, 21240 bytes data of original image with another BMP data (refer to Figure 7). | Image corrupted. | (Same with the original image) |
|                    | **Condition 3** |        | 184 KB     |
|                    |Adding data, example the 30 rows from another BMP data (refer to Figure 7) into the original data image. | Same like original image. | (more than original image) |

Based on Table 4, an experiment done by using 163 KB BMP image. The experiment has been tested by three conditions: Firstly, when the data deleted from file, the size is smaller than the original image and the image become cannot be displayed and error message is displayed instead. The second condition is replacing the data with another data and maintains equal size as the original image, hence the image will be corrupted but it can still be displayed. The last condition, the size is more than original image, the image is viewable but it looks same like the original image by adding another data.

The experiment shows certain important information regarding the behavior of a BMP file which can help in reconstruction of corrupted BMP image which is by changing the size affected the corruption of BMP image. From the first condition, it shows that if at least a byte of data is deleted from a BMP file, the file cannot be read by Photo Viewer because it is not a valid BMP file. The sizes of the BMP image are written in BMP Header based on the total of bytes in the image. In order to
view a BMP file with partial deleted data, the corrupted file must be in the same size as the original size stated in the header or the size in the header has to be changed according the current file size.

Second condition is replaced the data with another data in BMP file. Modifying or replacing the file with another file can cause corrupted in image. The file corrupted from the bottom to the top of the image because it was started the pixel from the bottom left corner in actual image.

While in third and last condition is added some of the bytes in BMP file. It is not change the original file. We can see the original file without any corrupted in image because the file can view only in range of file size. Modification of the file can be seen if change the file size and also adding the bytes.

5. Conclusion and Future Work
Changing of image size is one of the factors that can influence the corrupting of BMP image. In this paper, the experiment has been done by changing size of the BMP images through deleting, replacing and adding data. According to experiments, when the data has been deleted, the image becomes corrupted and non-viewable. While replacing the original data with another data, the image becomes corrupted but still can be displayed. In addition, the image can be displayed same like original image if another data has been added into the original data. It can be concluded that the size of the file determine the ability of the file to be displayed. BMP image is corrupted and cannot be displayed while deleting and replacing the file. Further investigation understanding about the technique that can be used on detecting corruption point is required to solve this issue. However, using information from this paper can help for investigating corruption point of BMP image which require to change the actual size of file to make the image viewable.

Acknowledgement
The authors would like to thank Ministry of Higher Education (MOHE), for granting RAGS Grant (Vote R066) to support this research. The authors would also like to thank Universiti Tun Hussein Onn Malaysia (UTHM) and Gates IT Solution Sdn. Bhd. for supporting this research. Thanks to anonymous reviewer for valuable comments.

References
[1] Zatyko, K. Defining Digital Forensics. 2007 [cited 2017 03/30/2017]; Available from: http://www.forensicmag.com/article/2007/01/commentary-defining-digital-forensics.
[2] Viraktamath, M.S. and G.V. Attimarad, Impact of Quantization Matrix on the Performance of JPEG. International Journal of Future Generation Communication and Networking, 2011. 4(3): p. 107-118.
[3] Cohen, M.I., Advanced carving techniques. Digital Investigation, 2007. 4(3): p. 119-128.
[4] 1stwebdesigner. Understanding the Most Popular Image File Types and Formats. 2016; Available from: https://1stwebdesigner.com/image-file-types/.
[5] Amitash. Between.net. Difference Between BMP and JPG. 2011 [cited 2017 03/8/2017]; Available from: http://www.differencebetween.net/technology/difference-between-bmp-and-jpg/.
[6] Yao, M., Color trapping on a halftoned bi-level bitmap. 2012, Google Patents.
[7] Elgabar, E.E.A. and H.A.A. Alamin, Comparison of LSB Steganography in GIF and BMP Images. International Journal of Soft Computing and Engineering (IJSCE) ISSN, 2013: p. 2231-2307.
[8] Por, L.Y., et al., An enhanced mechanism for image steganography using sequential colour cycle algorithm. Int. Arab J. Inf. Technol., 2013. 10(1): p. 51-60.
[9] Burger, W. and M.J. Burge, Digital image processing: an algorithmic introduction using Java. 2016: Springer.
[10] Chang, H.-W., Text character identification system and method thereof. 2012, Google Patents.
[11] Choi, C.-H., H.-Y. Lee, and H.-K. Lee, *Estimation of color modification in digital images by CFA pattern change*. Forensic science international, 2013. 226(1): p. 94-105.

[12] *Structure of BMP file*. [cited 2017 03/8/2017]; Available from: http://www.ue.eti.pg.gda.pl/fpgalab/zadania.spartan3/zad_vga_struktura_pliku_bmp_en.html.

[13] Poornima, R. and R. Iswarya, *An overview of digital image steganography*. International Journal of Computer Science and Engineering Survey, 2013. 4(1): p. 23.