Visualization of JPEG Metadata

Kamaruddin Malik Mohamad and Mustafa Mat Deris

Faculty of Information Technology and Multimedia,
Universiti Tun Hussein Onn Malaysia (UTHM),
86400 Parit Raja, Batu Pahat, Johor, Malaysia
{malik,mmustafa}@uthm.edu.my

Abstract. There are a lot of information embedded in JPEG image than just graphics. Visualization of its metadata would benefit digital forensic investigator to view embedded data including corrupted image where no graphics can be displayed in order to assist in evidence collection for cases such as child pornography or steganography. There are already available tools such as metadata readers, editors and extraction tools but mostly focusing on visualizing attribute information of JPEG Exif. However, none have been done to visualize metadata by consolidating markers summary, header structure, Huffman table and quantization table in a single program. In this paper, metadata visualization is done by developing a program that able to summarize all existing markers, header structure, Huffman table and quantization table in JPEG. The result shows that visualization of metadata helps viewing the hidden information within JPEG more easily.

Keywords: JPEG Metadata, Metadata Viewer, Digital Forensics.

1 Introduction

The International Organization for Standardization (ISO) started to look into ways to use high resolution graphics and pictures in computers in 1983[1]. Joint Photographic Experts Group (JPEG) was formed by the International Telegraph and Telephone Consultative Committee three years later to develop a standard procedure for encoding grayscale and color images. JPEG finally come out with guideline which is referred to ITU-T T.81 [2]. In 1992, JPEG File Interchange Format (JFIF) [3] is introduced with introduction of some metadata in the non entropy coded segment [4]. This is a de-facto file format, used for sharing in different applications and in the Internet [5]. JPEG Exchangeable Image File Format (Exif) was introduced by Japan Electronic Industry Development Association (JEIDA) in 1996, to be used for digital cameras [6, 7, 8]. Exif 2.2, a newer version of Exif was introduced by the Japan Electronics and Information Technology Industries Association (JEITA), which is formerly known as JEIDA.

There are four distinct modes of encoding processes namely, sequential DCT-based (also known as baseline JPEG), progressive DCT-based, lossless, and hierarchical [2]. Implementations are not required to provide all of these.

H. Badioze Zaman et al. (Eds.): IVIC 2009, LNCS 5857, pp. 543–550, 2009.

There are two alternatives of entropy coding namely Huffman coding and arithmetic coding. Patented arithmetic coding produces slightly better compression but slower than patent-free Huffman coding. Huffman coding is more widely used for JPEG image compression. No default values for Huffman tables (DHT) are specified, so that applications may choose tables appropriate for their own environments. However, default tables are defined for the arithmetic coding. Baseline JPEG uses Huffman coding, while the extended DCT-based and lossless processes may use either Huffman or arithmetic coding.

JPEG file are segmented by a special two-byte codes called markers. Most markers are start of marker segments containing a related group of parameters (e.g. DHT, quantization table (DQT), start-of-frame (SOF), define-arithmetic-coding (DAC), start-of-scan (SOS), define-number-of-line (DNL), define-restart-interval (DRI), define-hierarchical-progression (DHP), expand-reference-component (ERC), application segment (APP), reserved for JPEG extensions (JPG), comment (COM)); and some markers are stand alone (e.g. start-of-image (SOI), end-of-image (EOI), restart-interval-termination (RST)). All these markers are assigned two-byte codes, 0xFF byte followed by a byte which is not equal to 0x00 or 0xFF.

JFIF file must have "JFIF\0" identifier, without it maybe it is another type of JPEG files e.g. JPEG Exif [7] which is a commonly used format for digital camera that can be identified by a SOI and "Exif\0" identifier.

Formally, the Exif and JFIF standards are incompatible. This is because both specify that their particular application segment (APP0 for JFIF, APP1 for Exif) must be the first in the image file. In practice, many programs and digital cameras produce files with both application segments included. This will not affect the image decoding for most decoders, but poorly designed JFIF or Exif parsers may not recognize the file properly.

In this paper, a program is developed to enable visualization of JPEG metadata to include summary of markers, header structure, Huffman and quantization table. Thus, the software can be helpful to digital forensic investigator to compare photo obtained from the computer at the crime scene against the evidence photo. Furthermore, corrupted image can be reconstructed by refering to the evidence photo. Finally, the metadata is much easier to be viewed and understood as opposed to view it through hex editor.

The rest of the paper is organized as follows. Section 2 describes related works, section 3 discussed about the experiments done, section 4 discussed about the result and discussion and finally section 5 concludes this paper.

2 Related Works

There are mainly two types of software available in the Internet, the image viewer and the metadata viewer. In this paper, we are focusing on visualizing the metadata hidden in JPEG image. Most JPEG metadata viewers concentrate on visualizing the metadata content of JPEG Exif files only. An open-source Linux-based Exif metadata viewer program called jhead, allows the retrieval of Exif headers from JPEG Exif files [6]. The sample output from jhead is illustrated in Figure 1. The Windows-based Exif Image Viewer [9] output is displayed in Figure 2. More Exif metadata viewer software can be found at [10]. There are also metadata readers, editors and extraction tools available.

These softwares mainly focuses on visualizing typical selected Exif attributes information such as camera make, model, date and time, focal, exposure time, aperture as illustrated in Figure 1 and 2. However, we develop a more detailed program to visualize all available JPEG markers into summary, headers, Huffman tables and quantization tables.

```
File name: logo.jpg
File size: 800000 bytes
File date: 2009:01:01 09:09:09
Camera make: EASTMAN KODAK COMPANY
Camera model: KODAK LS443 ZOOM DIGITAL CAMERA
Date/Time: 2008:12:31 10:10:10
Resolution: 2448 x 1632
Flashed used: No
Focal length: 21.2mm
Exposure time: 0.011 s (1/90)
Aperture: f/4.7
Exposure: program (auto)
Jpeg process: baseline
```

Fig. 1. Typical jhead output

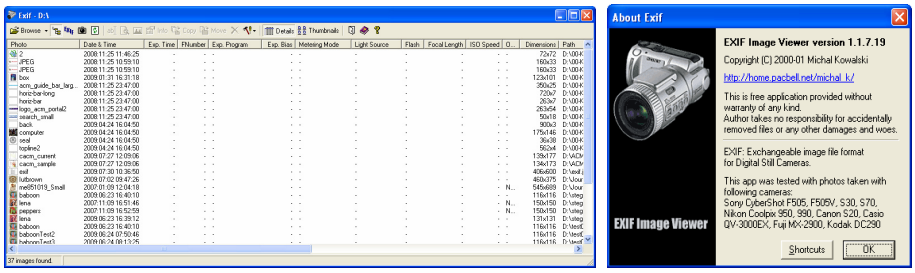

Fig. 2. Exif Image Viewer [9]

Table 1. Metadata readers, editors and extraction tools [10]

SOFTWARE	OPERATING SYSTEM
BR's EXIFextracter (freeware)	Windows
DateTree	Macintosh
EXIF Image Viewer and EXIF InfoTip (freeware)	Windows
Exif Reader (freeware)	Windows
EXIF Tag Parsing Library for Developers	
EXIF-O-Matic	Windows/ Macintosh /Linux
Exifer	Windows
EXIFread, EXIFren, and JPEGget	Windows
EXIFutils	Windows/ Macintosh /Linux
FotoTagger	Windows
Full Image Info	Windows
Ignore EXIF Color Space Plug-in for Photoshop	Macintosh
Ignore EXIF Color Space Utilities for Photoshop	Windows
iTag - Photo Tagging Software	Windows

Table 1.(*continued*)

Jhead	Windows/ Macintosh /Linux
Microsoft Photo Info	Windows
Namexif	Windows
Simple EXIF Viewer	Macintosh OS X
xMeta - Metadata Export from Photoshop	Macintosh

3 Experimentation

All valid two-byte markers found will be displayed on to the screen with its marker's name, its hex value and offset from the beginning of file. Stuffed byte (0x00 following 0xFF) and consecutive 0xFF will be skipped.

In this paper, the JPEG file will be visualized in few ways. It will be displayed as a summarized version by displaying all markers found in the file, specific JFIF metadata, specific Exif metadata, Huffman tables, quantization tables, and header. In this experiment, many JPEG files are used including "baboon" (baboon.jpg) and "lena" (lena.jpg). In this experiment, only results on baboon image will be discussed. Nevertheless, the result obtained could represent the general result for other JPEG images.

A summary of "baboon.jpg" is illustrated in Figure 4. A comprehensive list of markers are displayed including SOI, APP0, DQT, start-of-frame (SOF), DHT, start-of-scan (SOS) and EOI. From the summary, the file of size 4473 bytes is found to be baseline JFIF file with two DQT and four DHT. Moreover, the APPO marker (0xFFE0) used to identify JFIF file is located right after the SOI marker.

There are four Huffman tables used in baboon.jpg. Figure 5 illustrates the detail of first DHT DC table used with index 0x00. The first DHT segment size is 27 bytes (0x1B) including DHT length (2 bytes), DHT table index (1 byte), Huffman-bit-codes (16 bytes), and variable length DHT table (8 bytes for this table only). Huffman table detail is tabulate in Table 2.

A single byte of DHT table index is comprises of two values namely four most-significant-bit (MSB) table class (tc) and four least-significant-bit (LSB) destination identifier (th). tc of value 0x0 denotes AC table and value 0x1 denotes DC table. Thus, there are altogether two DHT DC tables (0x00, 0x01) and two DHT AC tables (0x10, 0x11).

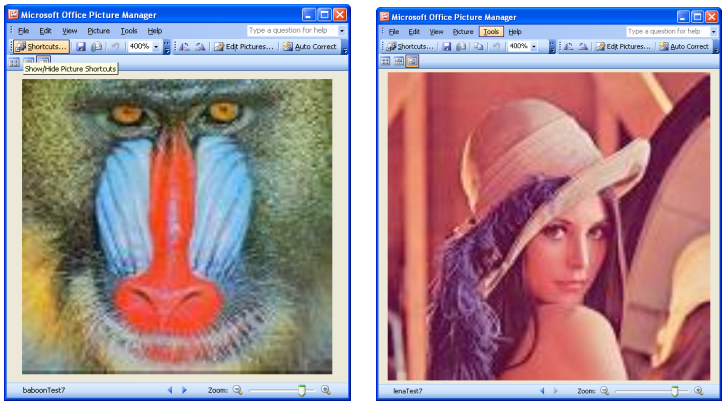

Fig. 3. Some of the test files used in this experiment; baboon and lena

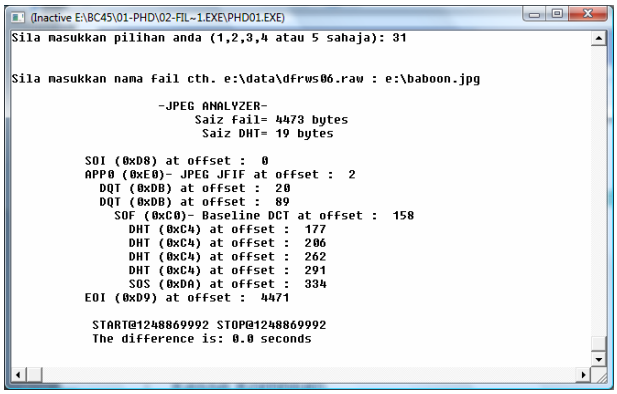

Fig. 4. A summary of baboon.jpg

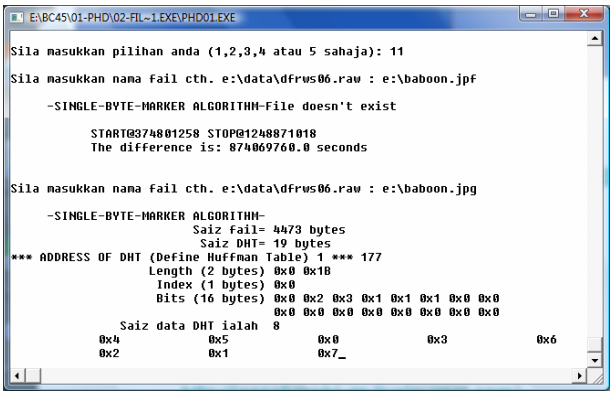

Fig. 5. First Huffman table details in baboon.jpg

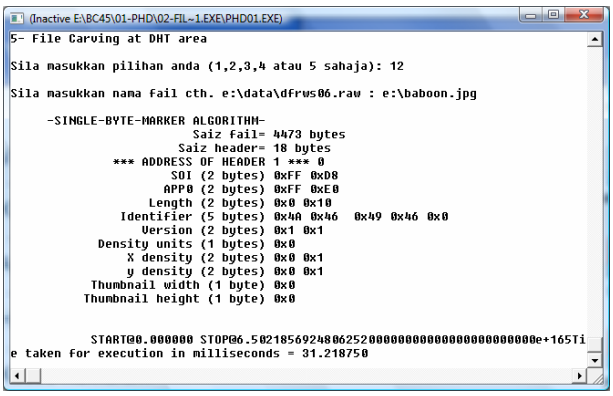

Fig. 6. Header details in baboon.jpg

Table 2. Detail of first Huffman table in baboon.jpg

Huffman-bit-code size (in bits)	Number of codes	Huffman-bit-code values
2	2	0x04, 0x05
3	3	0x00, 0x03, 0x06
4	1	0x02
5	1	0x01
6	1	0x07

The detail header for baboon.jpg is illustrated in Figure 6. The APP0 marker value is 0xFFE0 identifying JFIF file. Followed by size of header of 16 bytes (or 0x10 from 'header size' to 'thumbnail height'). JFIF identifier take up another 5 bytes ("JFIF" string terminated by a NULL ('0x00') or '0x4A 0x46 0x49 0x46 0x00'). Other values in the header including density unit (1 byte), x density (1 byte), y density (1 unit), thumbnail width (1 byte) and finally thumbnail height (1 byte).

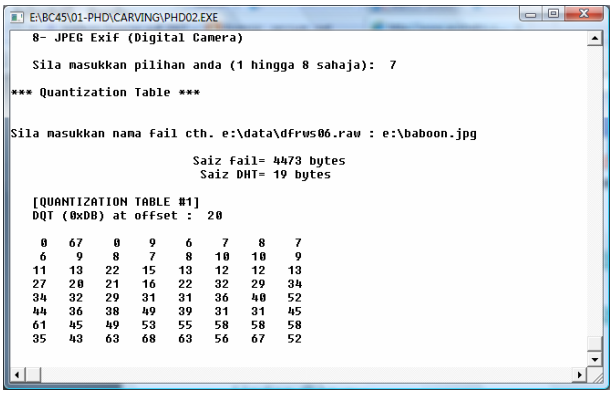

Fig. 7. First quantization table in baboon.jpg

There are two quantization tables in baboon.jpg. The first quantization table is illustrated in Figure 7. The quantization table is an 8x8 matrix with 64 values.

4 Result and Discussion

There are many standalone (e.g. SOI, EOI, RST) or start of segment two-byte markers (e.g. DHT, DQT, SOF, DAC, SOS, DNL, DRI, DHP, ERC, APP, JPG, COM). These markers are important information in digital forensic investigations. In this paper, these markers are visualized in many ways. First, all available markers are summarized with their offset address (refer to Figure 4). Secondly, information can be displayed by segment basis e.g. header (refer to Figure 5), DHT (refer to Figure 6), DQT (refer to Figure 7).

From the summary, digital forensic investigator can check similarity of the photo found in the computer at the crime scene with the evidence photo by looking at these markers and their offsets, comparison of Huffman and quantization tables. Two

similar images, but having different marker offsets, can mean that the photo has been altered. A corrupted image with one marker mising (e.g. EOI) can be rebuilt by adding the marker manually at similar offset. Thus, corrupted image (due to e.g missing marker, corrupted Huffman table, corrupted quantization table) can be rebuilt using data from the evidence photo. If both images are seen identical, but having different data in some parts of the files, this could also means that there is hidden message embedded in the image. Therefore, visualization of metadata definitely helps to reduce time taken for digital forensics investigator to gather evidence in criminal cases such child pornography and steganography.

5 Conclusion

There are a lot of information embedded in JPEG image than just graphics. Visualization of its metadata would benefit digital forensic investigator to view embedded data including corrupted image where no graphics can be displayed in order to assist in evidence collection for cases such as child pornography or steganography. The investigator can use these information to compare the photo obtained from the computer at the crime scene with the original photo that they kept as evidence. Comparison can be made by looking into Huffman tables, quantization tables and offsets of all markers found in the photo stored in the computer at the crime scene with the original evidence photo to see whether these images really identical in order to find out the perpertrator. If both images are seen identical, but having different data in some parts of the files, this could also means that there is hidden message embedded in the image. If the image is corrupted, it can be rebuilt using information from the metadata (e.g. missing EOI, missing few values from Huffman table etc.). There are already available tools such as metadata readers, editors and extraction tools but mostly focusing on visualizing attribute information of JPEG Exif. However, none have been done to visualize metadata by consolidating markers summary, header structure, Huffman table and quantization table in a single program. In this paper, metadata visualization is done by developing a program that able to summarize all existing markers, header structure, Huffman table and quantization table in JPEG. The result shows that visualization of metadata helps viewing the hidden information within JPEG more easily without having to use hex editor to look into its raw data.

Acknowledgement

This work was supported by Universiti Tun Hussein Onn Malaysia (UTHM).

References

1. Cohen, K.: Digital Still Camera Forensics. Small Scale Digital Device Forensics Journal 1(1), 1–8 (2007)
2. CCITT, Information Technology – Digital Compression and Coding of Continuous-Tone Still Images –Requirements and Guidelines, Recommendation, International Telecommunication Union, ITU T.81 (September 1992), http://www.w3.org/Graphics/JPEG/itu-t81.pdf

3. Wikipedia, http://en.wikipedia.org/wiki/JPEG_File_Interchange_Format
4. Hamilton, E.: JPEG File Interchange Format v 1.02, C-Cube Microsystems (September 1992), http://www.w3.org/Graphics/JPEG/jfif3.pdf
5. Swee, L.H.: JPEG for Digital Panel, Application Report SPRA664 , Texas Instrument (May 2000), http://focus.ti.com.cn/cn/lit/an/spra664/spra664.pdf
6. Alvarez, P.: Using Extended File Information (EXIF) File Headers in Digital Evidence Analysis. International Journal of Digital Evidence 2(3) (2004)
7. JEITA CP-3451, Exchangeable image file format for digital still cameras- Exif Version 2.2, Japan Electronics and Information Technology Industries Association (April 2002), http://www.exif.org/Exif2-2.pdf
8. Wikipedia, http://en.wikipedia.org/wiki/Exchangeable_image_file_format
9. WebAttack Inc., http://www.snapfiles.com/get/exifimageviewer.html
10. About.com: Graphics Software, http://graphicssoft.about.com/od/exifsoftware/EXIF_IPTC_XMP_Software_Metadata_Readers_Editors_Extraction_Tools.htm