

Analysis of DQT and DHT in JPEG Files

Nor Amira Nor Azhan

Department of Computer System and Communications

Faculty of Computing, Universiti Teknologi Malaysia

Skudai, Johor Bahru

namira5@live.utm.my

Ikuesan R. Adeyemi

Department of Computer System and Communications

Faculty of Computing, Universiti Teknologi Malaysia

Skudai, Johor Bahru

raikuesan2@live.utm.my

Shukor Abd Razak

Department of Computer System and Communications

Faculty of Computing, Universiti Teknologi Malaysia

Skudai, Johor Bahru

shukorar@utm.my

Abstract :

JPEG files are the most popular image format that is commonly analysed in image forensics research areas. The compositions of these files are prominent to help accomplish the problem of source identification, content classification, forgery detection, steganography, encryption, and file recovery. The purpose of this paper is to present the overview of JPEG file format, as well as identifying important characteristics of its content, which are related to signature and compression features. For analysis purposes, we used about 80 JPEG files produced from several digital devices to examine their contents. The results show that it is possible to differentiate images among JPEG files and from other file types, by examining specific content of JPEG files.

Keywords-component; *JPEG file; signature; compression*

I. Introduction

JPEG file standard, a file developed by the Joint Photographic Expert Group, is the commonly used image file format found in most electronic and storage devices [2-6]. Additionally, JPEG files are usually the default image format in almost all of digital cameras. Statistics reveals by global camera industries revealed that over \$55 billion digital camera sales on 2010 , while the figure is projected to increase to over \$65 billion by 2015 [7], thus making JPEG files the order of the digital world of photography.

80% to 90% of the criminal cases today have some kind of digital evidence [8] and one of such evidence is digital image file. Details analysis of JPEG content is critically required for image forensic research areas, since it is the widely used image format. The process of determining the device-source of an image is called image source identification. Image content classification and forgery detection are two sides of the forensic process. Image content classification analyzes image files for specific purpose, such as cyber pornography. On the other hand, image forgery detection field determines whether the images have been manipulated and altered. Image steganography is a technique that can be used to hide secret information in

image files while image encryption is a technique used to prevent the possible interpretation of an image file, so as to render such file unreadable to unintended audience. However, file recovery methods are needed to recover JPEG files that has been deleted or corrupted. Without the understanding of the underlying file structure, and bit positioning, it is almost impossible to recover a deleted or corrupted file irrespective of total or partial deletion process. This proves that the detail studies of JPEG file inner content's is necessary in order to help and solve specific research problems.

This paper discusses the basic information of JPEG files with special consideration on deep analysis of their internal structure. The next section begins with the discussions of basic information of JPEG files, their content and available tools that can be used with this file type. Furthermore, the paper details experiment and result of this process in Section 3, clarifying the important component contained in JPEG files. Concise conclusion elaborating current research challenges related to JPEG file image content analysis is also presented.

II. Background of Study

JPEG was created as a standard format to store photographic images. It only specifies how component values are stored but not the way colours need to be represented. JPEG File Interchange Format (JFIF), another acronym for JPEG, was introduced to allow images to be created and exchanged among different applications [9]. Usually, JPEG file has the extension of .jpg, .JPG, .jpeg or .JPEG.

Generally, JPEG file uses lossy compression methods that excludes some bits string (which is not detectable by the human eyes), before images are stored so as to minimize storage space as well as reduce file transmission time. For instance, to store 1 MB Windows BMP file, it requires only 50 KB with JPEG file [2]. There are four steps involve in compressing (or encoding) the JPEG file which are sampling, Discrete Cosine Transform, quantization and entropy coding. The reverse process is also true -decompression (or decoding). Miano (1999) elaborates more on Compressed Image File Format.

JPEG files are defined by four compression modes which are; hierarchical, progressive, sequential, and lossless. For file interchange process, the sequential compression method with Huffman baseline encoding [10] is strongly recommended to be used [9]. In sequential mode, images are encoded from top to bottom. Throughout this paper, discussions will focus more on baseline JPEG file.

A. State of the Art of JPEG File

The specification of JPEG file format or specifically markers can be in three different formats, which are JFIF (Figure 1), JFIF Exif (Figure 2) Exif (Figure 3). JFIF Exif is non-standard format in which both JFIF and Exif markers can be found in one image file [11].

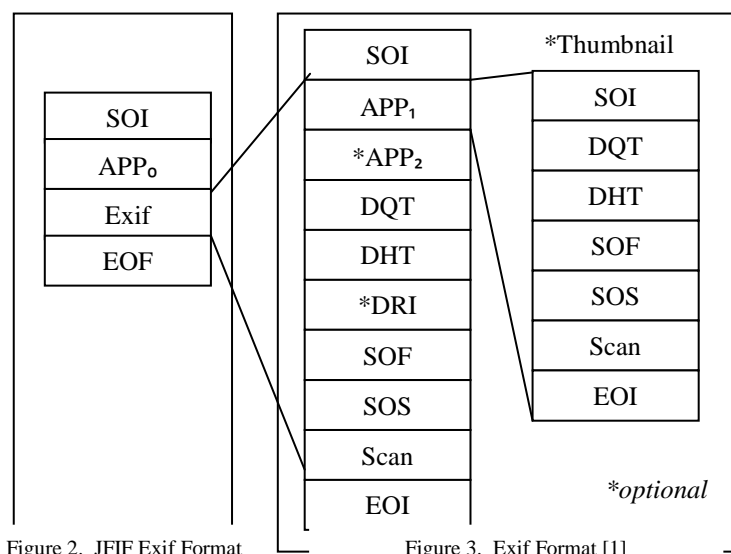
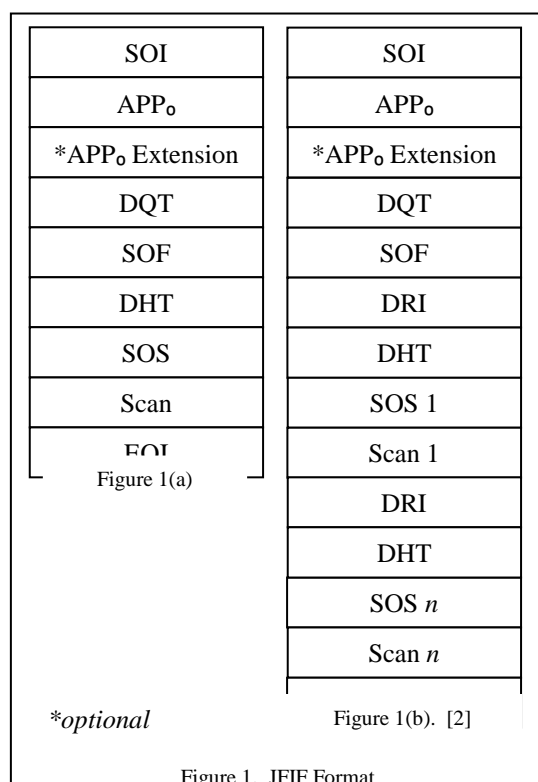


Figure 1, 2 and 3 shows JPEG markers contained in the file. Markers are used to specify the location of component in a file. 'Scan' in the figures signifies a compressed data in the file. JFIF format can be identified with APP₀ marker as shown in Figure 1(b). However, most of the JFIF file do not use the DRI marker, thus the format in Figure 1(a) is often used.

Exif format can be identified with APP₁ marker and sometimes with APP₂ marker (FlashPix extension data). Most of the Exif files are created with thumbnail image, usually without the DRI marker as shown in Figure 3. On the other hand, the JFIF Exif format contains two or three markers, APP₀, APP₁ and APP₂ which are incompatible as shown in Figure 2. In this case, the APP₀ marker comes without their extension and use the same Exif format. This specification is not realized by many applications and therefore working with this format might be a bit difficult.

Exif specifications allow one or two thumbnails to be embedded in the JPEG file [12]. Exif image specification is described in Figure 4. It shows that the locations of DQT (Define Quantization Table) and DHT (Define Huffman Table), are may be switched with each other depending on the devices that produced such image.

Single thumbnail		Two thumbnails	
Option 1	Option 2	Option 1	Option 2
SOI ₁	SOI ₁	SOI ₁	SOI ₁
APPX ₁	APPX ₁	APPX ₁	APPX ₁
SOI ₂	SOI ₂	SOI ₂	SOI ₂
*APPX ₂	DQT ₂ /DHT ₂	*APPX ₂	DQT ₂ /DHT ₂
DQT ₂	SOF ₂	DQT ₂	SOF ₂
SOF ₂	DHT ₂ /DQT ₂	SOF ₂	DHT ₂ /DQT ₂
DHT ₂	SOS ₂	DHT ₂	SOS ₂
SOS ₂	EOI ₂	SOS ₂	EOI ₂
EOI ₂	SOI ₃	EOI ₂	SOI ₃
DQT ₁	DQT ₃ /DHT ₃	SOI ₃	DQT ₃ /DHT ₃
SOF ₁	SOF ₃	*APPX ₃	SOF ₃
DHT ₁	DHT ₃ /DQT ₃	DQT ₃	DHT ₃ /DQT ₃
SOS ₁	SOS ₃	SOF ₃	SOS ₃
EOI ₁	EOI ₃	DHT ₃	EOI ₃
	DQT ₁	SOS ₃	DQT ₁
	SOF ₁	EOI ₃	SOF ₁
	DHT ₁	DQT ₁	DHT ₁
	SOS ₁	SOF ₁	SOS ₁
	EOI ₁	DHT ₁	EOI ₁
		SOS ₁	
		EOI ₁	

Figure 4. Image with Thumbnail/s Specifications

B. Tools

Exif format contain metadata or Exif data that describes the digital image itself. Basically, information like camera model used, creation time and date are usable for investigation purposes. For the sake of metadata viewer, there are several available tools that can be used as provided in Graphic Software web site [13]. Additionally, there are also approaches on how to identify only the JFIF header based on their markers [14-16]. Besides that, WinHex [17] and Hexeditor [18] are other tools that displays the actual content of JPEG files in hexadecimal form.

III. Analysis, Result and Discussions of JPEG Tables

DQT and DHT are important tables in JPEG files that allows the process of compression and decompression. There are several works done to display the contents of DQT and DHT. However, no significant exploration have been presented pertaining to the analyses of JPEG Metadata. JPEG Metadata Viewer [11] and JPEGsnoop [19] as open source tool that can be used for such purpose. These softwares synoptically displays the common information about the JPEG tables including markers, length of data and marker locations in a file.

This section will start with introduction of DQT and DHT, followed by examination of their contents. For analysis purposes, we used total of 80 JPEG files, in which each 40 files are generated from 20 digital cameras and mobile cameras. Each file was renamed from 1 to 80 names and the list of their associated devices is shown in Table 1. The files are image collection that has not undergone any modification. However, two images of every four images from each device (in no particular sequence), are rotated in order to obtain the proper display of images. In some cases, these rotated images exhibits some variance with the original image. This will be discussed further in next sub sections.

TABLE I. DISTRIBUTION OF DEVICES USED IN EXPERIMENT

Digital Cameras		Mobile Devices	
Device	File Name (.JPG/.jpg)	Device	File Name (.JPG/.jpg)
1. Canon EOS 450D	1-4	11. Samsung GT-I9300	41-44
2. Nikon Coolpix L16	5-8	12. Sony Ericsson K770i	45-48

3. Nikon D90	9-12	13. Sony Ericsson K850i	49-52
4. Olympus T105, T100, X36	13-16	14. Sony Ericsson W660i	53-56
5. Panasonic DMC-F2	17-20	15. Sony Ericsson W705	57-60
6. Ricoh R50	21-24	16. LG Electronics	61-64
7. Samsung ES60/SL105/ES63	25-28	GW520	65-68
8. Samsung S630	29-32	17. Nokia N73	69-72
9. Sony Cybershot	33-36	18. Nokia 1234	73-76
10. Sony DSC-W570	37-40	19. Nokia 5320	77-80
		20. Nokia 6233	

A. DQT

Quantization is a sort of matrices used by JPEG to control image compression. Every digital device and graphical application that produces JPEG files will have their own quantization table as implied in device setting. For instance, image can be set by low, medium or high quality and this mean users can choose the quality level to save the image.

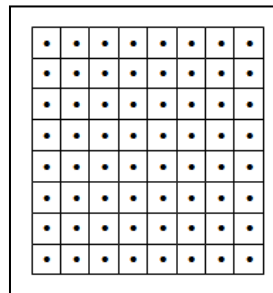


Figure 5. Image Pixels

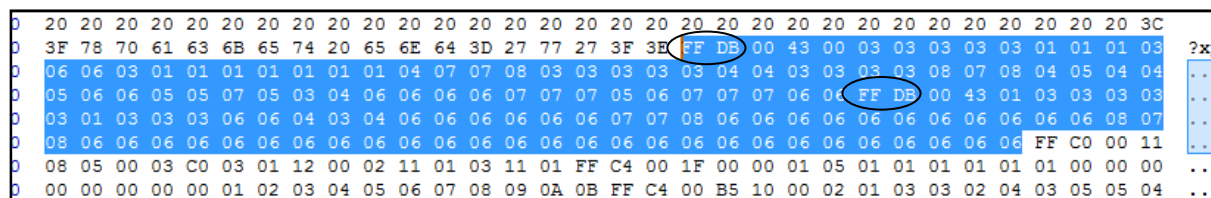
Image is a collection of information about very small dots or represented as pixel. In the first step of JPEG encoding, each pixel data is grouped into 8x8 blocks as shown in Figure 5. Each component is then converted from RGB (Red, Green, Blue) to YCbCr (Y: luminance, CbCr: chrominance) colorspace. Luminance represent brightness of the pixels, chrominance Cb specified blueness while chrominance Cr specified redness.

In the second step of JPEG encoding, DCT is used to transform each 8x8 block into a set of 64 DCT coefficients. The first value represents the DC coefficient and remaining 63 values are called AC coefficients. The third phase of JPEG encoding is to quantize the DCT coefficients by using 64 quantization values provided in DQT. Thus, the results will be used in the next encoding phase, which is entropy encoding.

Figure 6 shows the parameter specification of DQT. Basically, the value of length is calculated start with length field until the end of quantization values. From 40 images together with the 46 thumbnails of digital cameras, we found that 67 of them use 67-byte size of DQT, 54 of them use 132-byte size of DQT, and 4 of them use 197-byte size of DQT. From 40 images together with the 40 thumbnails of mobile cameras, we found that 80 of them use 67-byte size of DQT and 36 of them use 132-byte size of DQT. It shows that majority of the images use 67-byte size of DQT. A luminance table can be identified by 00 value while chrominance with 01 value. The quantization values are stored in zigzag order in JPEG files. Hexeditor tool is used to view the content of DQTs of one selected image as shown in Figure 7. Each DQT starts with FFDB marker and it shows that this image contains two DQTs.

Marker	Length	Y or CbCr Table	Quantization Values
FF DB	00 43	00 or 01
<i>2-byte</i>	<i>2-byte</i>	<i>1-byte</i>	<i>64-byte</i>

Figure 6. DQT Specification



The actual reordering of quantization values are obtained by converting the zigzag order as shown in Figure 8 by using JPEGsnoop [19].

```

*** Marker: DQT (xFFDB) ***
Define a Quantization Table.
OFFSET: 0x00005A92
Table length = 67
-----
Precision=8 bits
Destination ID=0 (Luminance)
DQT, Row #0: 3 3 1 1 1 1 1 3 4
DQT, Row #1: 3 3 1 1 1 1 3 4 6
DQT, Row #2: 3 3 1 1 3 3 5 6
DQT, Row #3: 6 3 1 3 3 4 5 6
DQT, Row #4: 6 4 3 3 4 5 6 7
DQT, Row #5: 7 8 3 5 7 6 7 7
DQT, Row #6: 7 8 4 5 6 7 7 7
DQT, Row #7: 7 8 3 4 5 6 6 6
Approx quality factor = 94.91 (scaling=10.18 variance=66.64)

*** Marker: DQT (xFFDB) ***
Define a Quantization Table.
OFFSET: 0x00005AD7
Table length = 67
-----
Precision=8 bits
Destination ID=1 (Chrominance)
DQT, Row #0: 3 3 1 3 6 6 6 6
DQT, Row #1: 3 3 3 4 6 6 6 6
DQT, Row #2: 3 3 3 6 6 6 6 6
DQT, Row #3: 6 4 6 6 6 6 6 6
DQT, Row #4: 6 6 6 6 6 6 6 6
DQT, Row #5: 7 8 6 6 6 6 6 6
DQT, Row #6: 7 8 6 6 6 6 6 6
DQT, Row #7: 7 8 6 6 6 6 6 6
Approx quality factor = 96.41 (scaling=7.18 variance=8.00)

```

Figure 8. Actual Ordering of Quantization Values

Refer to Figure 8, the precisions 8 bit represent the 8-bit per pixel sample luminance and chrominance. Quantization tables consist of luminance and chrominance matrices value. This means that the value of image brightness, blueness and redness have their compression ratio. JPEG may consists of two (one for Y, one for CbCr) or three DQTs (each for Y, Cb, Cr). Basically, luminance table become the first location of DQT, but there is a case where it becomes the second location of DQT. Table 2 shows the number of DQTs in 80 JPEG files.

TABLE II. NUMBER OF DQTS IN JPEG FILES

		#Camera Images	#Mobile Images
Image	YCbCr	2	3
	Y with default CbCr	-	1
Image with 1 Thumbnail	Image: YCbCr Thumbnail: YCbCr	14	16
	Image: YCbCr Thumbnail: Y with default CbCr	-	-
	Image: Y with default CbCr Thumbnail: YCbCr	-	5
	Image: Y with default CbCr Thumbnail: Y with default CbCr	16	11
Image with 2 Thumbnails	Image: YCbCr 1 st Thumbnail: YCbCr 2 nd Thumbnail: YCbCr	-	-
	Image: YCbCr 1 st Thumbnail: YCbCr 2 nd Thumbnail: Y with default CbCr	3	2
	Image: YCbCr 1 st Thumbnail: Y with default CbCr 2 nd Thumbnail: Y with default CbCr	1	-
	Image: Y with default CbCr 1 st Thumbnail: Y with default CbCr 2 nd Thumbnail: Y with default CbCr	4	2
	Image: Y with default CbCr 1 st Thumbnail: Y with default CbCr 2 nd Thumbnail: YCbCr	-	-
	Image: Y with default CbCr 1 st Thumbnail: YCbCr 2 nd Thumbnail: YCbCr	-	-
Total:		40	40

Result in Table 2 shows that only five images do not have thumbnail and the rest of images may contain either one or two thumbnails. Majority of images (14 images from cameras, 16 from mobile) are provided with luminance and chrominance DQT tables. In the second highest number of images (16 images from cameras, 11 from mobile), whenever luminance table is provided, the default chrominance table is used. The default table values are different depending on the devices and thus give the various quality levels of images.

In this study, JPEG was saved at 90%, with 10% of the data has loss. Although the quality values has been identified for several digital cameras [20], the quantization table in images may not match that quality values. Therefore, the last quality saved in the JPEG can be identified by approximating the quality value based on DQT. The following figures shows the result of approximated quality value for 80 JPEG files by using a method developed by Neal Krawetz [21].



Results from experiment shows that even if DQT is missing and there is requirement to view the images, it is possible to use DQT from other available image [4]. As a result, the quality of the image will be different, either higher or lower, but image still can be viewed. However, even if one byte data from DQT is modified and changed, the image become corrupted file and cannot be viewed. This type of error can be found in one of dataset available in the Internet [22], in which one byte of DQT length field has been changed in a JPEG file.

Other mechanism to control JPEG compression is entropy encoding. There are two standard ways to compress JPEG file either by using Huffman encoding or arithmetic coding. DHT is important to obtain successful decoding and it is provided to eliminate the zero numbers in 64 quantized coefficients by applying variable length codes. Thus, shorter codes will be assigned to more frequently used bits. Figure 11 shows the parameter specification of DHT.

Marker	DHT Length	Table Class	Table Destination	Huffman Code	Values of Huffman code
<p>This papers was presented at 2nd International Conference on Information Integration & Computing Application (ICIIA2013) . Singapore, on May 4 – 5 , 2013 Page ... 8.....</p>					

FF C4	0 or 1	0 or 1
2-byte	2-byte	4-bit	4-bit	16-byte	No fix length

Figure 11. DHT Specification

Refer to Figure 11, again, the value of length is calculated start with length field until the end of Huffman code values. From 80 images being analyzed, it shows that all of the Y-DC table use either 31-byte or 418-byte size of DHT, Y-AC and CbCr-AC table use 181-byte size of DHT, CbCr-DC table use 31-byte size of DHT.

Hexeditor tool is used to view the content of DHTs of one selected image as shown in Figure 12. Each DHT in that image starts with FFC4 marker and thus this image contains four DHTs.

08	05	00	03	C0	03	01	12	00	02	11	01	03	11	01	FF	C4	00	1F	00	00	01	05	01	01	01	01	01	01	00	00	00
00	00	00	00	00	01	02	03	04	05	06	07	08	09	0A	0B	FF	C4	00	B5	10	00	02	01	03	03	02	04	03	05	05	04
04	00	00	01	7D	01	02	03	00	04	11	05	12	21	31	41	06	13	51	61	07	22	71	14	32	81	91	A1	08	23	42	B1
C1	15	52	D1	F0	24	33	62	72	82	09	0A	16	17	18	19	1A	25	26	27	28	29	2A	34	35	36	37	38	39	3A	43	44
45	46	47	48	49	4A	53	54	55	56	57	58	59	5A	63	64	65	66	67	68	69	6A	73	74	75	76	77	78	79	7A	83	84
85	86	87	88	89	8A	92	93	94	95	96	97	98	99	9A	A2	A3	A4	A5	A6	A7	A8	A9	AA	B2	B3	B4	B5	B6	B7	B8	B9
BA	C2	C3	C4	C5	C6	C7	C8	C9	CA	D2	D3	D4	D5	D6	D7	D8	D9	DA	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	F1	F2	F3
F4	F5	F6	F7	F8	F9	FA	FF	C4	00	1F	01	00	03	01	01	01	01	01	01	01	01	01	01	00	00	00	00	00	01	02	03
04	05	06	07	08	09	0A	0B	FF	C4	00	B5	11	00	02	01	02	04	04	03	04	07	05	04	04	00	01	02	77	00	01	02
03	11	04	05	21	31	06	12	41	51	07	61	71	13	22	32	81	08	14	42	91	A1	B1	C1	09	23	33	52	F0	15	62	72
D1	0A	16	24	34	E1	25	F1	17	18	19	1A	26	27	28	29	2A	35	36	37	38	39	3A	43	44	45	46	47	48	49	4A	53
54	55	56	57	58	59	5A	63	64	65	66	67	68	69	6A	73	74	75	76	77	78	79	7A	82	83	84	85	86	87	88	89	8A
92	93	94	95	96	97	98	99	9A	A2	A3	A4	A5	A6	A7	A8	A9	AA	B2	B3	B4	B5	B6	B7	B8	B9	BA	C2	C3	C4	C5	C6
C7	C8	C9	CA	D2	D3	D4	D5	D6	D7	D8	D9	DA	E2	E3	E4	E5	E6	E7	E8	E9	EA	F2	F3	F4	F5	F6	F7	F8	F9	FA	FF

Figure 12. DHTs in JPEG File

There are four DHT tables that specified in table class and definition field. They are Y-DC, Y-AC, CbCr-DC and CbCr-AC tables. DQTs and DHTs are related with each other in order to further compress the data.

The observation shows that:

- If either images or thumbnails are using DQT of Y with default CbCr, then they are using DHT of 2 DC tables
- If either images or thumbnails are using DQT of YCbCr, then they are using DHT of 2 DC and 2 AC tables

Refer to Table 2, it shows that there are higher number of images that use DQT of Y with default CbCr (27 images) and DQT of YCbCr (30 images). Thus, the two situations described above are quite important, since a total of 57 out of 80 images matches with this situation. As a result, there are also higher number of images that consist of 2 DC tables, and images with 2 DC and AC tables. On the other hand, emphasizing on JFIF files, 100 of this type of files uses 2 DC and AC tables [23]. Additionally, emphasizing on Exif files, 69 out of 76 files uses 2 DC tables [24].

IV. Discussion

There are several tools/approaches for viewing the important components (DQT and DHT) in JPEG file, from which valuable forensic information can be extracted. Considering DQT for instance, the image quality value can be estimated, while number of information about the used device can also be revealed. The analysis presented in this paper, shows that some devices produce similar image quality value. However, they also exhibit some characteristics which can be used to highlight the default quality level of the device. If the default quality levels for the images are known, then the number of resaved image can be determined. This

findings therefore provides useful insight for image modification analysis, as well as to image source identification.

As shown in the analysis in section 3, the number of DQT is related to DHT. Thus, images can be concluded to use 2 DC tables if they are using DQT of Y with default CbCr; otherwise, 2 DC and 2 AC tables, if they use DQT of CbCr. Thus, from this result, the length of the DHT can be estimated to depend on the DQTs that constitute the image. However, the length cannot be estimated with accurate number because the Huffman code values are not fixed. However, with this study of specific format of DQT and DHT, it is useful in differentiating JPEG file from other file types.

Additionally, in comparison with previous research works, this research focuses on one type of JPEG file comprising various JPEG formats such as Exif and JFIF Exif. It is worthwhile to mention that the JFIF Exif format is not recognized as standard JPEG format. However, this format is usually ignored by most of the research work, while we found that most of the rotated images used this type of format. From the analysis, we found that, due to the effect of image rotation, for both digital and mobile cameras, the un-rotated image (original image form) uses Exif specification, while rotated images uses JFIF Exif specification.

V. Conclusion

This paper discussed an overview of JPEG file format, as well as identifying important characteristic of JPEG files. While being an important component for compression and decompression, valuable information (evidence) can also be extracted from DQT and DHT. We used about 40 JPEG files from digital cameras and 40 JPEG files from mobile cameras to examine the image content. From the examination of DQT, the approximated quality value of images can be calculated depending on various devices that are used to produce JPEG files. On the other hand, the relation between DQT and DHT can be used to differentiate between JPEG files from other file types as well as differentiating between JPEG files themselves.

REFERENCES

- [1] JEIDA, "Digital Still Camera Image File Format Standard (Exif) Version 1.0," 1995.
- [2] J. Miano, *Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP*: Association for Computing Machinery, Inc. (ACM), 1999.
- [3] M. I. Cohen, "Title," unpublished].
- [4] H. T. Sencar and N. Memon, "Identification and Recovery of JPEG Files with Missing Fragments," presented at the Digital Forensic Research Workshop, 2009.
- [5] M. Xu and S. Dong. 2009, Reassembling the Fragmented JPEG Images Based on Sequential Pixel Prediction.
- [6] J. v. d. Bos and T. v. d. Storm, "Bringing Domain-Specific Languages to Digital Forensics," *ICSE '11*, 2011.
- [7] reportlinker.com. 2012, *Market Research Reports*. Available: <http://www.reportlinker.com/ci02061/Camera.html>
- [8] K. Medaris and R. Mislan. 2008, *Expert: Digital evidence just as important as dna in solving crimes*. Available: <http://news.uns.purdue.edu/x/2008a/080425T-MislanPhones.html>
- [9] E. Hamilton, "JPEG File Interchange Format Version 1.02 " 1992.

- [10] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the I.R.E.*, pp. 1098–1102, 1952.
- [11] K. M. Mohamad and M. M. Deris, "Visualization of JPEG Metadata," *IVIC 2009, LNCS 5857*, pp. 543–550, 2009.
- [12] M. M. Deris, *et al.*, "Carving JPEG Images and Thumbnails Using Image Pattern Matching," *2011 IEEE Symposium on Computers & Informatics*, pp. 78-83, 2011.
- [13] About.com. 2013, *Graphics Software*. Available: http://graphicssoft.about.com/od/exifsoftware/EXIF_IPTC_XMP
- [14] K. M. Mohamad and M. M. Deris, "Single-byte-marker for Detecting JPEG JFIF Header using FORIMAGE-JPEG," *2009 Fifth International Joint Conference on INC, IMS and IDC*, pp. 1693-1698, 2009.
- [15] K. M. Mohamad, *et al.*, "Dual-Byte-Marker Algorithm for Detecting JFIF Header," *ISA 2010, CCIS 76*, pp. 17 – 26, 2010.
- [16] K. M. Mohamad, *et al.*, "Detecting JFIF Header using FORHEADER," *International Journal of Security and Its Applications*, vol. 5, pp. 23-36, 2011.
- [17] X.-W. S. T. AG, "WinHex," 2012.
- [18] M. Horz, "Hexeditor," 2009.
- [19] C. Hass. 2012, *JPEGsnoop*. Available: www.impulseadventure.com/photo/
- [20] C. Hass. 2008, *Comparing JPEG Quality*. Available: <http://www.impulseadventure.com/photo/jpeg-quality.html>
- [21] N. Krawetz. 2005, *jpegquality.c*. Available: <http://www.hackerfactor.com/src/jpegquality.c>
- [22] B. Carrier. 2005, *Digital Forensics Tool Testing Images* Available: <http://dftt.sourceforge.net/>
- [23] K. M. Mohamad and M. M. Deris. 2009, Fragmentation Point Detection of JPEG Images at DHT Using Validator. 173-180.
- [24] M. Karresand and N. Shahmehri, "Reassembly of Fragmented JPEG Images Containing Restart Markers," presented at the European Conference on Computer Network Defense, 2008.