Journal of Software Engineering and Applications, **2013, 6, 62-66**
doi:10.4236/jsea.2013.63b014 Published Online March 2013 (http://www.scirp.org/journal/jsea)

Carving Thumbnail/s and Embedded JPEG Files Using Image Pattern Matching

Nurul Azma Abdullah, Rosziati Ibrahim, Kamaruddin Malik Mohamad

Faculty of Computer Science & Information Technology, UTHM, Malaysia.
Email: {azma, rosziati, malik}@uthm.edu.my

Received 2013

ABSTRACT

Images (typically JPEG) are used as evidence against cyber perpetrators. Typically the file is carved using standard patterns. Many concentrate on carving JPEG files and overlook the important of thumbnail in assisting forensic investigation. However, a new unique pattern is used to detect thumbnail/s and embedded JPEG file. This paper is to introduce a tool call PattrecCarv to recognize thumbnail/s or embedded JPEG files using unique hex patterns (UHP). A tool called PattrecCarv is developed to automatically carve thumbnail/s and embedded JPEG files using DFRWS 2006 and DFRWS 2007 datasets. The tool successfully recovers 11.5% more thumbnails and embedded JPEG files than PredClus.

Keywords: Component; JPEG Image; Pattern Matching; File Carving; DFRWS 2006/07; Thumbnail

1. Introduction

A method to recover files from the disk collected from the crime scene which is known as cyber evidences is called file carving [1-3]. Year by year, with the increasing number of computers and other digital devices usages, file carving technique also evolve drastically. In carving a JPEG images, the focus is in fragmentation file carving. Although there are researchers discussed on fragmentation such as [2-4], it is not yet completely solved.

This paper is introducing a way to assist in reducing number of JPEG files to be processed for fragmentation point detection. A JPEG image can be embedded into other file types such as doc, ppt, etc. Furthermore, thumbnails can be embedded into a JPEG image itself to ease the recovering and organizing of the original image. A JPEG image can contains none, single or two thumbnails in the image itself. A thumbnail which is a reduced version of an image carried similar feature as the original. This thumbnail is always mistaken as another JPEG image. Therefore, knowledge of thumbnail's existence helps investigator to separate JPEG files with thumbnail/s and concentrates to investigate correct point where the real fragmentation occurs. They can then identify which JPEG image is fragmented with another JPEG images. With this way, they can ascertain that those fragments are belongs to two or more different files, not file/s within a file. This is important because during the reassembling process, if a thumbnail is mistakenly identified as another JPEG files, the original file may corrupt because of missing fragments. Consequently, this is also accelerating the reassembling process by allowing investigators to concentrate on real fragmentation situation.

In this paper, a novel algorithm called PattrecCarv is proposed. PattrecCarv is developed to recover thumbnail/s or embedded files from DFRWS 2006 and 2007 datasets using unique hex patterns (UHP). The output can be used as a pre-processing data to simplify the process for recovering of JPEG image.

The rest of the paper is organized as follows. Section II is the related works consists of an overview of JPEG standard and thumbnail and embedded JPEG files while Section III brief on PredClus algorithm. Section IV describes the proposed PattrecCarv algorithm. Section V describes the experimentations done. Section VI describes the result and discussion. Finally section VII concludes this paper.

2. Related Works

2.1. An Overview of JPEG Standard

Computer forensics is to recover evidences resides on a computer, by mean to solve pornography cases [1-3]. This involves image files obtained from the perpetrator in certain format like Bitmap and JPEG but most common format is JPEG. JPEG is popular because of its compressed file that can reduce the size required to allocate an image. Joint Photographic Experts Group (JPEG) was formed by International Telegraph and Telephone

Consultative Committee in 1986 inspired by an effort of International Organization of Standard (ISO) to find ways to use high resolution graphics and pictures in computers [4]. JPEG introduced compression standard for both grayscale and color continuous-tone images. The details of JPEG compressed data formats can be found in [5]. There are two types of JPEG that are mostly used today, JPEG File Interchange Format (JFIF) and JPEG Exchangeable Image File Format (Exif) [6,7]. JFIF is popular for internet file while EXIF is the popular image file format used for digital camera [8].

2.2. Thumbnail and Embedded JPEG Files

Both in JFIF and Exif format allow for embedding thumbnail/s into a JPEG file. A JPEG image with a complete SOI/EOI can be embedded into an original JPEG image to ease the recovering and organizing of the original image. This file is known as thumbnail. Thumbnails are reduced size version of images that can be used to recover and organize the picture [9] while embedded JPEG files are referred to original JPEG files that are embedded to other types of files such as PPT, WORDS and EXCEL. Thumbnails are used to speed up images search or page load on the Internet and also being used in image organizing programs. Thumbnails are compatible on most modern operating systems or desktop environments such as Microsoft Windows, Mac OS X, KDE and GNOME [10]. A JPEG image can contain none or a single or two thumbnails. Therefore, a JPEG image can have several SOI/EOI pairs [11]. Mohamad in [12] and [13] asserted the role of thumbnail to serve as a method of recognizing the corrupted images because of its small size that have a better chance for full recovery without corruption [14]. A thumbnail carried similar features as the original. Hence, using thumbnail/s, crime investigators can identify which images or pictures that have potential to be used as evidences against cyber perpetrator.

Guo in [9] proposed thumbnails as a method to recover JPEG image from fragment data. In brief, thumbnails do serve multiple roles. Besides contributing in the process of recovering and organizing JPEG files, thumbnails help in recognizing corrupted images and also, information about thumbnail's location can be used in carving fragmentation JPEG images to recover the original files. Abdullah et al. [15] proposed PredClus as a method to recognize thumbnail/s and embedded JPEG files. However, using PredClus which using cluster size to determine the location of thumbnail/s or embedded file may miss some thumbnails that resides at the start of cluster. This situation occurs when a JPEG image with thumbnail/s require more than one cluster to store the data. Sometimes, the start of thumbnail will be at the start of second cluster. In this situation, the thumbnail/s will be ignored by PredClus. Hence, an alternative technique to distinguish thumbnail or embedded JPEG file with the original is by using pattern matching technique. In carving JPEG images especially fragmented JPEG files, it will ease the process of preparing evidence if the carver can distinguish between original images, thumbnails and embedded images.

3. Predclus Algorithm

PredClus is developed to automatically determine cluster size of a dataset. Using this information, JPEG images that are not located at the starting address of any cluster are marked as thumbnails or embedded JPEG files. The algorithm of PredClus is introduced to predict cluster size used in both DFRWS 2006 and 2007.

First, data from dataset is read. These data are in hex values. The hex values then matched with the standard JPEG header. However, in this experiment, additional markers are also used instead of standard JPEG header, 0xFFD8 alone. The additional markers used are 0xFFE0, 0xFFE1, 0xFFE2, 0xFFC4 and 0xFFDB. When matched, the offset for each markers matched is retrieved. Using formula as mentioned earlier, the determinant value is calculated. If the determinant value = 0, then file found is counted. This is done for each cluster size which are 512-byte, 1-kb, 2-kb, 4-kb and 8-kb cluster. Please refer to [15] for detail explanation of PredClus.

After the determinant value for all JPEG files in the datasets is extracted, files found for each cluster size then are summed. The percentage for each cluster size is calculated. Then, a report is produced.

4. Pattreccarv Algorithm

This section discusses on the development of the proposed algorithm called PattrecCarv. The algorithm is adapted from dual-byte-marker algorithm proposed by [1] to detect JPEG headers (SOI), thumbnails and embedded JPEG files. DFRWS 2006 and 2007 datasets are used for testing this algorithm. Nevertheless, the algorithm can also work with other datasets.

A thumbnail in JFIF format can be recognized using UHP in **Table 1** while a thumbnail in EXIF format as shown in **Table 2**. On the other hand, embedded JPEG files can be recognized using UHP as in **Table 3**.

Table 1. The first 17 bytes of JFIF thumbnail.

SOI	APP0	Length	Identifier					Version	Additional Markers
0XFFD8	0XFFE0	0X00	0x4A J	46 F	49 I	46 F	00 NULL	0x0102	0x0100480048
2 bytes	2 bytes	1 byte	5 bytes					2 bytes	5 bytes

Table 2. The first 6 bytes of non-JFIF thumbnail.

SOI	Validated markers	
0xFFD8	0xFFDB/0xFFC4	0x0084
2 bytes	2 bytes	2 bytes

Table 3. The first 14 bytes of embedded JPEG files.

SOI	APP0	Length	Identifier	Version	Additional Markers
0XFFD8	0XFFE0	0X00	0x4A 46 49 46 00 J F I F NULL	0x0102	0x0003
2 bytes	2 bytes	1 byte	5 bytes	2 bytes	5 bytes

Basically, the PattrecCarv algorithm works as follows:

STEP 1: Identify start-of-image (SOI or 0xFFD8) markers (refer to **Table 1**)

- If two-byte structure read is a SOI marker, then jump to the 9th hex value.

STEP 2: Once SOI is found, locate the embedded UHP (refer to Table III)

- If the 9th hex value is the embedded UHP, then embedded JPEG header is found.
- Else, read the next value.

STEP 3: If the read hex values are the UHP for thumbnail (refer to **Table 1** & **Table 2**), then thumbnail is found.

Repeat STEP 1, STEP 2 and STEP 3 until end of data.

The algorithm of PattrecCarv is illustrated in **Figure 1**.

5. Experimentation

This section discusses on the experiment designed for ThumbedCarv model as illustrated in **Figure 2**. Both algorithms, PredClus and PattrecCarv are installed into this model and the results from both algorithms are compared. Both algorithms are developed using C++ language in Windows 7 with Intel® Core TM2 Quad CPU and 2GB of physical memory.

The input of this model is from two datasets, DFRWS 2006 and DFRWS 2007. Comparisons are made for these algorithms (PredClus and PattrecCarv) based on the number of successfully JPEG thumbnails and embedded JPEG files recovered.

The details of PredClus algorithm are discussed in [15]. PattrecCarv consists of function to carve thumbnails and embedded JPEG files using pattern recognition technique. To carve thumbnails, three sets of validated markers as shown in **Table 1** and **Table 2** are used while validated markers for carving embedded JPEG files are shown in **Table 3**.

```
1.  Read data image
2.  Initialize hex_values
3.  Initialize thumb_markers
4.  Initialize embedded_markers
5.  Find JPEG header
6.  If found
7.      Jump to 9th hex_values
8.      If hex_values== embedded_markers
9.          Read CurrentOffset
10.         Save the CurrentOffset
11.     Else
12.         Read next hex_values
13.         If hex_values==thumb_markers
14.             Go to the most recent SOI
15.             Save the CurrentOffset (location of SOI)
16.         Endif
17.     Endif
18. If not end of data image, repeat step 5
19. Generate report
```

Figure 1. Algorithm used in PattrecCarv for carving thumbnails and embedded JPEG files.

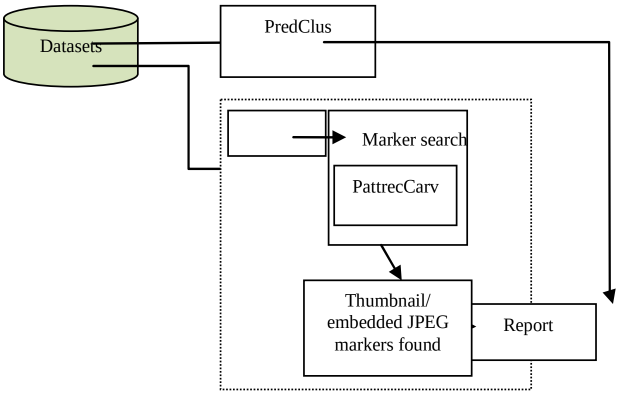


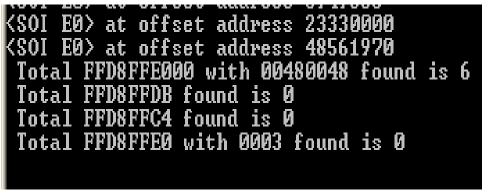
Figure 2. ThumbedCarv model.

The algorithm starts with reading the dataset. Once JPEG SOI markers are found, the next APP0 markers are read. If it is matched, it reads the next 9th hex value. If next two bytes hex values match the embedded JPEG file markers, the current offset is recorded. If not, next one byte hex value is read. Once, a JFIF thumbnails markers are detected, the offset value of the thumbnail is recorded. If the APP0 markers are not found after SOI markers, the algorithm checks the next two bytes hex values. If they match with validated markers as in **Table 1**, then thumbnail is detected. Finally, the report of thumbnails and embedded JPEG files is generated.

6. Result and Discussion

The screenshot for PattrecCarv output can be clearly examined in **Figure 3** and **Figure 4**. **Figure 4** depicts total number of thumbnails with JFIF headers detected from DFRWS 2006 dataset is 6 and none for embedded JPEG files. There are also no thumbnails using UHP 0 x FFD8 **Figure 4** shows total number of thumbnails and embedded JPEG files detected is 33 which is correspond with 1 thumbnail with the JFIF header, 18 thumbnails recognized using UHP of 0xFFD8 and 0xFFDB, 2 thumbnails detected using UHP of 0xFFD8 and 0xFFC4 and 12 embedded JPEG files.

Table 4 shows the comparisons done on PredClus and PattrecCarv algorithms. From the table, there is not a distinct difference of execution time but there is some interesting findings in term of thumbnail/ embedded file found, original detect as thumbnail, thumbnail or embedded JPEG files missed and false detection.



```
<SOI E0> at offset address 23330000
<SOI E0> at offset address 48561970
 Total FFD8FFE000 with 00480048 found is 6
 Total FFD8FFDB found is 0
 Total FFD8FFC4 found is 0
 Total FFD8FFE0 with 0003 found is 0
```

Figure 3. Screenshot of PattrecCarv output using DFRWS 2006 dataset.

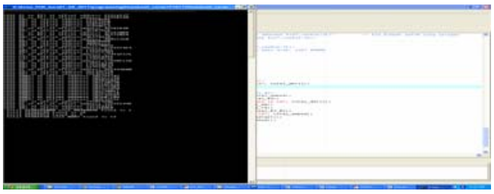


Figure 4. Screenshot of PattrecCarv output using DFRWS 2007 dataset.

Table 4. Comparison done on PredClus and PattrecCarv.

	Experiment 1 (PredClus)		**Experiment 2 (PattrecCarv)**	
Dataset	Dataset 2006	Dataset 2007	Dataset 2006	Dataset 2007
Thumbnail/ embedded file found	5	23	5	26
Thumbnail/ embedded file that is missed	0	3	0	0
Original detected as thumbnail	0	0	0	4
False detection	0	0	1	3
Time taken (sec)	3.0	27.0	4.0	27.0

Clearly, PattrecCarv detects more thumbnail/embedded file compared to PredClus though it falsely detect 4 files as thumbnails. PredClus is using cluster size information to determine the detected file is either thumbnail or embedded file or original file. That is the reason it did not make any mistake in determining thumbnail/embedded file. However, using PredClus, some thumbnails can be missed. This is caused by a big JPEG file that pushes the header of thumbnail to be stored at the start of file. Furthermore, PredClus cannot differentiate between thumbnails and embedded files because it does not know any information about the file detected; only the size of cluster is known. Although PattrecCarv has falsely detected 5 files, but it does not miss to carve any thumbnails/embedded files and separate between thumbnails and embedded files. The original file detected as thumbnail is caused by fragmented data. An experiment has been conducted manually to investigate the cause of this condition. It is found that all original files detected as thumbnails are fragmented with another JPEG files or other files.

7. Conclusion

JPEG file can be in a form of original JPEG file, thumbnail or embedded in another file. However, the importance of thumbnail and embedded file should not be overlooked in forensic investigation. This paper introduces a unique file pattern matching technique, which is embedded in a tool called PattrecCarv. PredClus assumes that all images starting from the first byte of a cluster as an image which may mistakenly detect a thumbnail as an original file where as PattrecCarv uses unique file pattern matching technique. Based on experiments done using DFRWS 2006 and 2007 data sets, PattrecCarv successfully carves thumbails and embedded JPEG files more efficiently as compared to PredClus.

8. Acknowledgment

The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) for supporting this research.

REFERENCES

[1] S. L. Garfinkel, "Digital forensic research: the next 10 years," *Digital Investigation*, *7*(1), 2010, S64-S73.

[2] A. Pal and N. Memon, "Automated reassembly of the file fragmented images using greedy algorithms," *IEEE Trans. Image Processing*, 15(2), 2003, 385-393.

[3] M. Karresand and N. Shahmehri, "Reassembly of fragmented jpeg images containing restart markers," in *2008 European conference on computer network defense*.

[4] M. I. Cohen, " Advanced carving techniques," *Digital Investigation*, 4(1-4), 2007, pp. 119-128

[5] The International Telegraph and Telephone Consultative

Committee (CCITT) 1992 “Information technology—digital compression and coding of continuous-tone still images–requirements and guideline (ITU-T T.81),” 1992. Retrieved Sept. 5, 2012, from World Wide Web Consortium (W3C): http://www.w3.org/Graphics/JPEG/itu-t81.pdf

[6] K. M. Mohamad, and M. Mat Deris, “Visualization of JPEG metadata,” in: *Proceeding. of the 2009 first International Visual Informatics Conference on Visual Informatics*.

[7] E. Hamilton, “JPEG file interchange file format version 1.02.” Retrieved Sept. 5, 2012 from JPEG Committee Homepage: http://www.jpeg.org/public/jfif.pdf

[8] P. Alvarez, “ Using extended file information (exif) file headers in digital evidence analysis,” *International Journal of Digital Evidence*. 2(3), 2004.

[9] H. Guo and M. Xu, “A method for recovering jpeg files based on thumbnail,” in: *Automation and Systems Engineering (CASE) 2011 International Conference.*1-4.

[10] Thumbnail. Retrieved Sept 5, 2012, from Wikipedia: http://en.wikipedia.org/wiki/Thumbnail.

[11] A. Merola, “Data carving concepts,” 2008. Retrieved Sept. 5, 2012,

[12] from SANS Institute: http://www.sans.org/reading_room/whitepapers/forensics/datacarving-concepts_32969Y.

[13] K. M. Mohamad, A. Patel and M. Mat Deris, “Carving JPEG images and thumbnails using image pattern matching,” in *2011 IEEE Symposium on Computers and Informatics*.

[14] K. M. Mohamad, A.Patel, T. Herawan and M. Mat Deris, “myKarve: JPEG image and thumbnail carver,” *Journal of Digital Forensic Practice*. 3, 2011, 74-97.

[15] K. Cohen, “Digital still camera forensics. *Small Scale Digital Device Forensics,”* 1(1), 2007, 1-8.

[16] N.A. Abdullah, R. Ibrahim,and K.M. Mohamad,”Cluster size determination using JPEG files,” in *Proceedings of the 12th international conference on Computational Science and Its Applications*.

[17] K.M. Mohamad, T.Herawan and M. Mat Deris, “Dual-byte-marker algorithm for detecting JFIF header.,”in Bandyopadyay, S. M., Adi, W., Kim, T. & Xiao, Y. (eds.) *Information Security and Assuranc,*. 17-26, 2010,Springer,Heidelber.