Title	Analyzing storage media of digital camera
Author(s)	Tse, KWH; Chow, KP; Law, FYW; Ieong, RSC; Kwan, MYK; Tse, H; Lai, PKY
Citation	The 2009 International Workshop on Forensics for Future Generation Communication Environments (F2GC-09) in conjunction with CSA 2009, Jeju Island, Korea, 10-12 December 2009. In Proceedings of CSA, 2009, p. 1-7
Issued Date	2009
URL	http://hdl.handle.net/10722/125689
Rights	Proceedings of the International Conference on Computer Science and Its Applications. Copyright © IEEE.

Analyzing Storage Media of Digital Camera

Kenneth W.H. Tse, K.P. Chow, Frank Y.W. Law, Ricci S.C. Ieong,
Michael Y.K. Kwan, Hayson Tse, and Pierre K.Y. Lai
Department of Computer Science
The University of Hong Kong
{whktse, chow, ywlaw, scrieong, ykkwan, hkstse, kylai}@cs.hku.hk

***Abstract*— Digital photography has become popular in recent years. Photographs have become common tools for people to record every tiny parts of their daily life. By analyzing the storage media of a digital camera, crime investigators may extract a lot of useful information to reconstruct the events. In this work, we will discuss a few approaches in analyzing these kinds of storage media of digital cameras. A hypothetical crime case will be used as case study for demonstration of concepts.**

***Keywords*—**storage media, FAT, file system analysis, time analysis

I. INTRODUCTION

Digital photography has become popular in recent years. With the portability and the decrease in cost of photo-taking-capable devices, people are able to take digital photos anytime and anywhere. As a result, photographs have become common tools for people to record every tiny parts of their daily life.

Digital photographs are often involved in crime investigation too. Photos containing traces to the criminals may be found on the Internet, or a flash memory card seized from the suspect's home may prove or disprove certain hypothesis.

Unlike traditional film photography, digital photographs can provide a lot more useful information (often known as metadata) than their image content only. If the photos are found on storage media, there could be even more traces left behind. This information could help a lot in crime investigation, in particular helping in reconstructing the events occurred.

In this work, we focus on the analysis of storage media of digital cameras. We are going to discuss a few approaches to reconstruct events from these media. A hypothetical crime case will be used as case study for demonstration of concepts.

II. RELATED WORKS

There are not many previous works focusing on the analysis of storage media of digital camera, nor on FAT file system.

However, timestamps analysis has been a hot topic in the research. Chow et al., has analyzed the file timestamps on NTFS file system to develop heuristics rules on behavior characteristics of related digital files [1]. Willassen has also work on discovering traces on antedating by developing causality reasoning on sequence number and allocation sequence on a NTFS file system [2].

III. BASIC CONCEPTS

In this section, we will introduce briefly on some basic concepts related to the storage media of digital cameras in the following subsections.

A. Storage Media

Digital cameras and other hand help devices use flash memory cards as primary storage. There are a few commonly used storage media card types for digital cameras on the markets. The majority of consumer digital cameras usually use Secure Digital card (SD cards) [3], while some others use alternatives like Memory Stick and xD-Picture Card. Others photo-taking-capable devices like mobile phones and PDA use uses variant of SD card like mini-SD and macro-SD cards. In professional digital camera, CompactFlash cards are commonly used. All these types of flash memory cards in the market can have capacities of a few gigabytes, or even more than one hundred gigabytes for some of them.

Although forensics on the physical layer of the storage media may be possible, this would not be the focus of our work.

B. File System

Most of the flash memory cards use File Allocation Table (FAT) [4] as file system. The FAT file system is named after its file allocation table at the beginning of the file system. The FAT stores files and directories entries as linked list. Clusters of the file blocks are linked one after another.

When there are enough space, files are usually stored in continuous clusters. However, after some files are deleted, discontinued clusters of space may be created. When a new file is written to the file system, it may span across several discontinued clusters, and it is now fragmented. Fragmentation causes performance drop and thus is not preferred. FAT itself has no implementation on preventing fragmentation. However, the implementation of that file system driver may include mechanisms to avoid. This will be further discussed in later sections of this work.

The two common FAT used nowadays are FAT16 and FAT32. The major difference is the length of cluster field and thus resulting in different limits in capacities.

Although FAT has been replaced by other more advanced file systems like NTFS [5] in computers, it is still widely used in flash memory cards due to the straight forward design and

978-1-4244-4946-0/09/$25.00 ©2009 IEEE

the operation system portability of the file system. As a result, research effort is still worth on this seemingly old generation of file system.

C. Structure and File Naming

Due to the fact that a memory card of the digital cameras may be plugged into other devices, most of the digital cameras follow a standard named Design rule for Camera File system (DCF) [6] for better file management. The standard was developed by Japan Electronics and Information Technology Industries Association (JEITA). The current version of the standard is 2.0. Folder structure and file name scheme were defined in the standard. In short, the memory card for digital camera contains a folder named "DCIM", which stands for "Digital Camera Image". Inside the folder, there may be at most 900 sub-folders inside and named in the format "###ABCDE", where "###" is a unique number 100 to 999 and "ABCDE" can be any five free characters allowed in the standard, usually they are used to represent the manufacturer name. These folders are called DCF directories and may contain DCF objects which are mostly image files and their thumbnails. The naming convention for DCF objects is "ABCD####", where "ABCD" can be any four free characters allowed in the standard and "####" are number from "000"1 to "9999", plus the file extension.

D. EXIF Data

In addition to DCF, the JEITA also defined the Exchangeable image file format (EXIF) specification [7], which is commonly used to stored metadata in image file formats like JPEG and TIFF. The latest version of the EXIF is 2.21.

The EXIF data usually exists in the beginning of the image file. It may contain information such as camera manufacturer, camera model, camera settings, copyright information, and etc. Information which interests digital investigator most would be the date and time the photo was taken. In some digital camera, a unique ID of the camera or a special copyrighted message inputted by the user can be stamped automatically to each photo taken by the camera. Moreover, GPS coordinates may also be included in EXIF if such information is available and supported by the camera. All these information could be helpful in an investigation.

IV. METHODOLOGY

In this work, we will mainly focus on the analysis of digital camera storage media in the file system level, using time stamps as well as its metadata. We will also suggest some other approaches which may be useful in case-specific manner.

Before going into details of the approaches, we would like to highlight some goals for the analysis. The major goal is to reconstruct the file-related events occurred on the storage media through extracting information or pattern of it. We are interested to see the order of file created, when or even how the files are accessed, or deleted. Our approaches are of two types, one for finding out absolute date time of the occurrences of event, while another type for finding out the relative sequence of occurrence. Both of the types would be helpful in rebuilding the story. In the following sections we would discuss a few useful approaches which may be used together to result in a higher chance of success reconstruction. When approaches for finding absolute time and relative order are used together, we may also place the image files at correct location on the timeline, or even detect potential antedating.

A. Analyzing Timestamps

The first approach to put events into order is straight forward – by using timestamps associated with the files. Timestamps are also a useful indicator to show the exact time for activities performed on a file. Basically there are two major sources of timestamps, one from the file system while another from the metadata of the image files.

In FAT file system, each file is stamped with 3 timestamps, namely last modified time, last access time and creation time. The three time stamps are usually referred as the MAC time of a file. By experiments and observations, the MAC time patterns of files on FAT file system share similar properties of those in NTFS file system as discussed in Chow's paper [1]. For example, when a file is copied from another drive, only the creation time will be changed while the modified time would remain unchanged. In other words, if certain image files have last modify time earlier than creation time, they are likely copied from other drive onto the file system. It may be sign of abnormal activities in which user copied certain image back to the camera.

When analyzing the FAT file system timestamp, we should, however, note that the timestamps on FAT file system are rounded to the nearest two seconds [8].

As mentioned earlier, another source of timestamps is from the metadata of the image file. In most of the cases, this metadata refer to EXIF data, which is widely adopted by most photo-taking-capable digital devices. According to the EXIF specification, there are a few EXIF tags storing date and time information. The "DateTime" (0x132) tag stores the date and time the file was changed; the "DateTimeOriginal" (0x9003) tag stores the date and time when the image was originally created (i.e. when it is taken); and the "DateTimeDigitized" (0x9004) tag stores the date and time when the image was digitalized. In the case of digital camera, these three tags should be the identical if the image is left untouched. They should also have the same value as the last modify time of the image file at file system level. If any of these value different, it may be an indicator of changes in the image file.

It should be noted that most of analysis approach using timestamps share similar problems that the accuracy of the timestamps relies heavily on the accuracy of the system clock. There may also be problems on the timestamps caused by misconfiguration of time zone settings. An exceptional case to this is that when the information is extracted from the "GPSDateStamp" (0x1d) and "GPSTimeStamp" (0x7) tags in the EXIF data. The two tags exist when the device are GPS-capable and have the "Geotagging" feature turned on. They indicate the capture date and time information as UTC (Coordinated Universal Time) received directly from the GPS satellite, which should have high degree of trust level. If such

tags exist, they should help the investigator a lot in anchoring the image files on a timeline.

B. Analyzing File System Sector Allocation

Another approach of analysis is using the file system sector allocation pattern, which could help in finding relative occurrence sequence. Before performing such analysis, we should first deduce the sector allocation scheme. However, the FAT standard has not defined such a scheme. In other words, the allocation scheme would depend on the implementation of the file system driver. We performed experiments attempting to find out the common schemes used in different implementations.

In the experiment, we used a small size SD card having capacity of 32MB. We create files on the memory card and observe the allocation clusters using a free tool named DiskView [9] developed by Bryce Cogswell. Several snapshots were taken during the experiment in order to observe the changes. The experiment was repeated on each setup to reduce the opportunity for result caused by randomness. The experiment procedures are as follow:

1. Create 10 image files of similar sizes on a fleshly formatted memory card.
2. Delete 2 images files in the middle of the list.
3. Create an image whose size was smaller than the 2 deleted images.
4. Create a complete sector dump of the memory disk as restore point.
5. Repeat step 3 with another small size file.
6. Recover the memory card to the restore point created at step 4.
7. Create a large image file, whose size is able to be fit in the trailing empty spaces, but not in the space in between.

We first execute the experiment on a consumer digital camera. The camera model used in this experiment is Canon Digital IXUS 860 IS. In this setup, image files were created by taking photographs of random objects. The list of files used and their file size was included in Table A-1 in the appendix. The media was removed from the camera and plugged into a computer for taking snapshot using DiskView tool. A few snapshots were taken as illustrated in Fig. 1.

Each of the snapshots consists of two major visualizations. The one on the top shows all the clusters on the file system. Each of the boxes represents a single cluster and filled with different color, blue represent occupied clusters, white represent free clusters while yellow represent clusters allocated for certain specified file. The lower visualization shows areas of non-fragmented (in blue), fragmented (in red) and free clusters (in white) in the file system.

We can deduce from Fig. 1(a) that the files are allocated one after another when they are created in sequence. When some files was deleted, empty spaces would be created in the middle of the allocated sectors as illustrated in Fig. 1(b). When new files of small size, which can be fitted into the empty space in between, were created, they will be placed in the beginning of the free cluster area as illustrated in Fig. 1(c) and (d). When a file whose size is larger than the middle free cluster area, a portion of it will be used to fill in the free area, while the remaining portion will be put in the beginning of the trailing free spaces (thus the file was fragmented) as illustrated in Fig. 1(e). In conclusion, the allocation scheme in the camera's implementation was a first fit algorithm which just

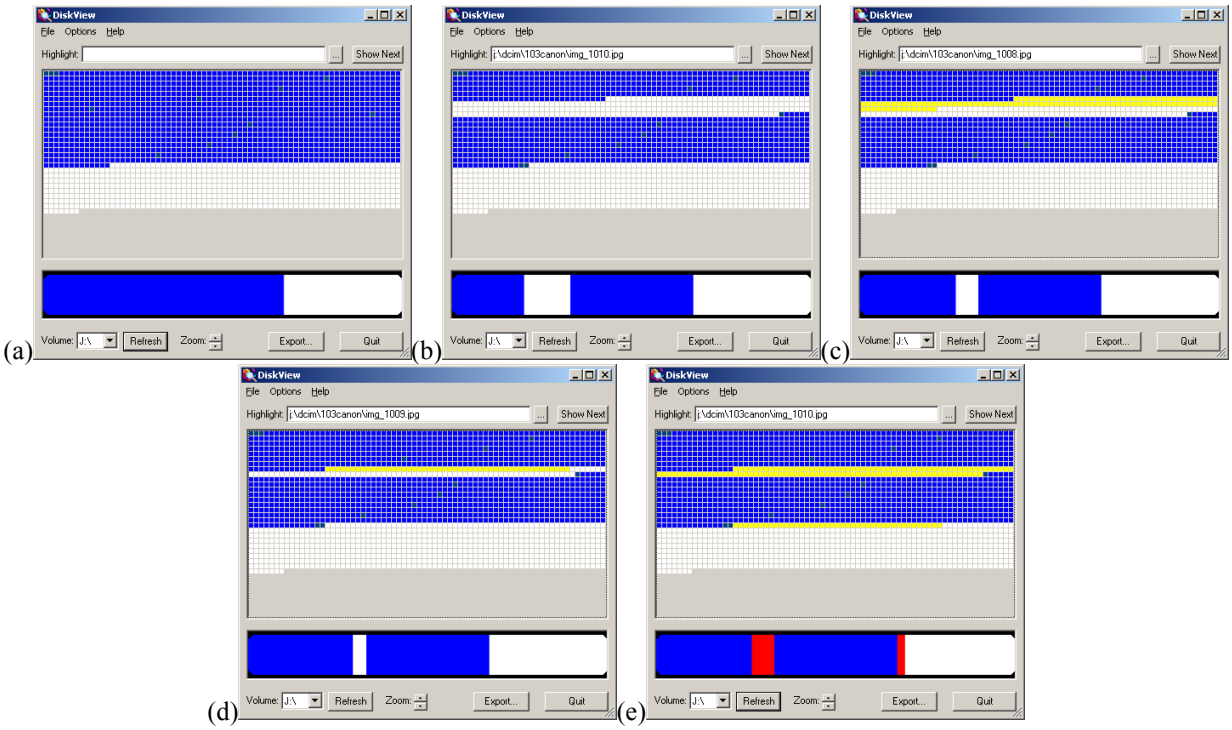

Figure 1. Snapshot taken after (a) step 1; (b) step 2; (c) step 3; (d) step 5; and (e) step 7 in the first setup

put the file in the first available clusters in the file system.

The experiment was then performed on the same SD card mounted to a computer running Windows XP Service Pack 3. In this setup, image files were created by copying existed files from the computer to the SD card. The same set of files created in previous setup was used. The resulted snapshots in this setup were identical to that of the previous one except that of step 7 as illustrated in Fig. 2.

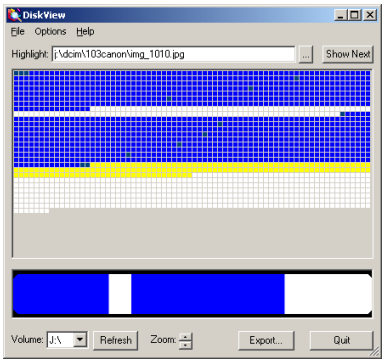

Figure 2. Snapshot taken after step 7 in the second setup.

It was observed from Fig. 2 that the big file did not fill the middle empty space. It was also observed in the lower visualization that all the files were not fragmented. We may deduce that a best-fit algorithm, which tries to put the file in the first space fitting the whole file, if available, was used in current setup, attempting to avoid fragmentation.

In the above experiments, two different allocation schemes were observed. Thus, it is recommended to perform tests similar to above on case specific setup, if known, to deduce the allocation scheme used before performing analysis on storage media. After determining the allocation scheme, hypothesis may be built up to test against the allocation scheme in order to find out the order of events occurred on the file system.

C. *Analyzing Filenames*

The third analysis approaches would be using the special filename patterns used on the digital camera storage as defined in the DCF. This may be helpful in finding relative order of photograph taking. As described in Section 1.3, the file name of each DCF directories and objects shall contain 3 and 4 digits of number respectively. Although the DCF standard did not define the sequence of digit used and left the flexibility to the camera manufacturer, it is commonly observed that those numbers are assigned in increasing basis. Thus they can be potentially used as sequence numbers for deducing the order.

However, similar to the case of file system sector allocation, different assignment scheme may be used in different devices. The major bias exists in the assignment after some files are deleted. In most digital camera, the next assignment number in the filename would not be affected by deleting existed files. That is, the camera either finds the largest number in filename, increase that by 1 to form the new number, or they store the last assigned number counter in their internal memory and update after each shot. Some other devices, for example, a Nokia 6300 mobile phone, would use a first fit algorithm to find the first available number to be used in newly created image. It should also be noted that on some cameras, the user is allowed to manually reset the file numbering to initial settings, or instruct the camera to automatically reset for each different storage media. As a result, the investigator should perform test on the case-specific setup to deduce the assignment scheme used before performing analysis with filename.

D. *Other Approaches*

There are other techniques available for analyzing the digital camera storage media. For example, in some more advanced digital camera, the number use shutter open will be recorded as the "shutter count" value. Owing to the scalability of the EXIF standard to allow manufacturer to include custom information, some camera may stamp the shutter count to each of the images. Since the shutter count is strictly increasing for each camera, it can be a nice indicator for the relative order of taking of the photos. However, such information is usually encoded in the "Makernote" (0x927c) tag which is in proprietary formats defined by the camera manufacturers.

There may be other approaches applicable in case-specific manner. For example, in some of the cases the image content can be good indicator of order. For examples, images of clock may tell you the time when the photograph was taken, the direction and angle of sunshine in the image may be correlated with the geographical information to estimate the photo-taking time. The states of the existing objects or even the non-existence of objects in the different images may also be compared to deduce a relative sequence of them. All these are all case specific and would require the investigator to discover.

V. A CASE STUDY

In this section, we attempt to apply our discussed approaches on a hypothetical case for demonstration purpose.

A. *Background*

A man X took obscene photographs of his ex-girlfriend Y, using a digital camera, and blackmailed her by threatening to publish the photographs on the Internet. X was later arrested by the police and a SD card was seized from his possession. The SD card was analyzed and found to contain 5 deleted obscene pictures of Y. The accuracy of the photographic evidence was challenged due to some inconsistency between the timestamps and Y's statement. According to Y, the photographs were taken under duress on 5-6 Jan 2009. X then copied these photographs to his computer and blackmailed her.

Only the list of the files on the SD card, related metadata such as timestamps, file sizes, 1st sector numbers and EXIF (for images) were preserved for our analysis. Using this information, we will try to reconstruct the chorographical order of the photographs. We will also attempt to find out the exact date and time the photographs were taken.

B. *Detailed Analysis*

We started the analysis by looking the file system, file types and folder structure on the SD card. The card was using a

FAT32 file system. A "DCIM" folder was found containing the deleted photographs. Other folders such as "ringtones", "games" and etc, were also discovered. Varies types of files such as music, video and executable files were found in these folders. This indicated that the user not only used the memory card in the digital camera, but also in other devices such as mobile phone and PC. The investigation become more complex as the timestamps may be stamped by different clocks. Moreover, since many of them are recovered files, the time stamp information was not completed – not all the files had all of the three MAC time. The creation and last modified time of the files range from Jan 2007 to Jul 2008, while those of the photographs were mainly on Jan 2008. It was still the same case when we looked into the timestamps stored inside the EXIF data of the photographs. This seemed to be contradictory with the statement of Y. We tried to explain this phenomenon by a clock reset hypothesis. It was observed that, when some of the digital cameras went out of battery, the system clock would be reset to 1 Jan 2008. As the timestamps of the photographs were very likely to be stamped by the digital camera, if the clock of the X's camera was reset some times before the incident, the observed phenomenon could be possible.

In order to have another view of the files on the file system, we decided to visualize them using modified version of candle chart. We obtain the start sector number from the list and calculate an approximate range of allocated sectors using its file size. The range of each file was then presented by a candle in the chart. Candles for images file were colored in red while the others were colored in blue. Due to the limitation of space, only portion of the chart was illustrated in Fig. 3.

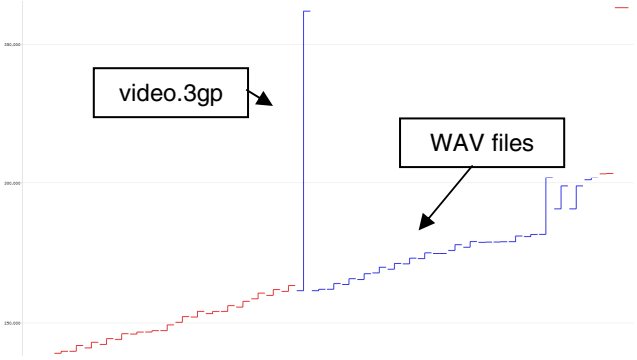

Figure 3. Estimated allocated sector range for some files on the SD card

The Y-axis of the chart indicates the sector number while the X-axis are the list of files sorted from left to right by its start sector number. It can be easily noticed from the chart that there are some overlapping of files. It could be explained as there may be rounding error during the conversion of file size units. The overlapping of the file video.3gp and the following WAV files, was however strange that most of these files still exist on the media card. A possible explanation to may be that there are high degree on fragmentation on the file system. The video.3gp, which was large compared to other files, may be quite seriously fragmented and the error in the estimated allocation range could be notable. This could be confirmed if we could have access to the FAT table of the file system to extract the allocated sectors of the files for analysis.

On the other hand, we tried to analyze the file allocation pattern on the file system together with the file timestamps. We focused only on the recovered and carved photographs for simplicity. They are listed in Table I in the appendix. The last modified times of the carved files were filled according to file name assigned by the carver. Due to the potential high degree of fragmentation, we only took the 1st sector of the file into consideration in this attempt. We plot the 1st sector of each file against their file timestamp. Each file was represented as a cross on the plot. Recovered images are marked as red, while carved images are marked as blue. They are illustrated in Fig. 4.

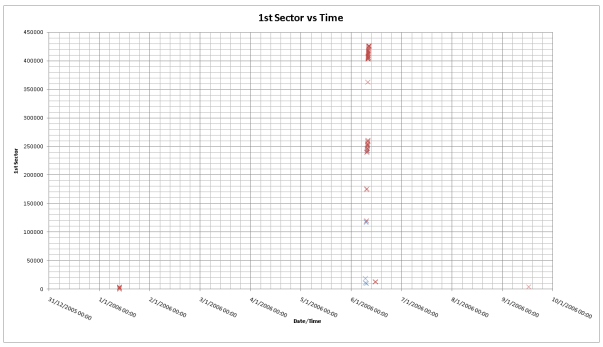

Figure 4. Plot showing the 1st sector to modified time relation of recovered/carved photographs.

From the plot, if files with close last modified time were put into the same group, 3 distinct groups can be formed with three separate dates, 1 Jan 2008, 6 Jan 2008 and 9 Jan 2008. The batch number was also marked in Table II in the appendix. By observing only the 6 Jan 2008 group, as illustrated in Fig. 5, an increasing trend could be observed.

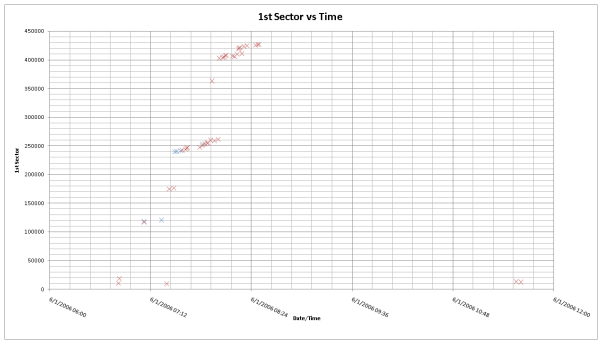

Figure 5. Plot showing the 1st sector to modified time relation of recovered/carved photographs in the 6 Jan 2008 group.

The few exception points which do not fit into the increasing trend could be explained by the hypothesis that some earlier files were deleted within the time slot the series of photos were captured, and the file system driver is using first fit algorithm on allocating sectors for new file as described in section 3.2. We can also estimate the filenames of the carved files in the batch by the hypothesis that the camera created new filenames by increase the largest existed one by 1. However, this hypothesis would not be further elaborated in this work. By

relating the last access time of photos in this batch with other files on the SD card, it seemed that someone deleted this batch of files on 5 Nov 2008, in order to free some spaces for putting new files including games, and other media files onto the SD on 6 Nov 2008. Since the moving of files involved other device (probably a computer) whose clock was not affected by the clock reset hypothesis on the digital camera and was believed to be accurate, these timestamps could be regarded as trustworthy. As a result, this batch of files was not related to the case and could be excluded.

Although photographs in the batches on 1 Jan 2008 and 9 Jan 2008 have close start sector number and a seemingly matched sequence number in the filename, they could not be in the same batch because "?XYZ0006.JPG" came before "?XYZ0005.JPG" and seemed to be overwritten completely. Despite the chance that it was a result of fragmented files, the order of first sectors of the two files would contradicts with the filename hypothesis we have during the analysis of 2nd batch. In other words, the 5 images in the 1st batch should be the focus of the investigation.

By examining the last access time of the 5 images in the 1st batch, we can observed that all of them had been last accessed on 6 Jan 2009, which matched the statement of Y that X copied files to the computer after taking photographs of her on 5-6 Jan 2009.

C. *Results*

By applying approaches suggested in section IV which analyze the timestamps on file, EXIF data together with the file sector allocation and sequence number in the photograph's filename, we successfully set up clock reset hypothesis and clear all mysteries on the timestamps of files on the SD card. The accuracy of the photographic evidence has also been justified by our hypothesis, which matches the statement of Y.

VI. CONCLUSIONS

To conclude, we have discussed a few approaches on analysis data, especially photographs on the storage media, which usually uses FAT-based file system, of a digital camera, or similar photo-taking-capable devices. We also demonstrated the approaches using a hypothetical case and successfully verified the results.

However, during the analysis, we also observed the potential effect of fragmentation on file system on our discussed approaches. Further research on minimizing the impact of fragmentation to the analysis could be performed. In our work, there only a few different setups were been used in the experiments thus we are not able to develop more generic rules or a behavior database when using the discussed approaches on analysis. In addition, we observed the opportunities on developing automatic tools to help better visualizing timestamps, allocated sectors, and other form of sequence number (such as that in the filename) of files.

VII. APPENDIX

TABLE I. LIST OF FILES USED IN THE ALLOCATION PATTERN EXPERIMENT.

Filename	Size (bytes)	Remark
IMG_0998.JPG	2,002,106	
IMG_0999.JPG	2,144,400	
IMG_1000.JPG	2,018,074	
IMG_1001.JPG	1,943,858	deleted in step 2
IMG_1002.JPG	2,046,855	deleted in step 2
IMG_1003.JPG	1,898,857	
IMG_1004.JPG	2,228,291	
IMG_1005.JPG	2,205,223	
IMG_1006.JPG	2,122,791	
IMG_1007.JPG	2,143,226	
IMG_1008.JPG	2,043,468	created in step 3
IMG_1009.JPG	781,972	created in step 5
IMG_1010.JPG	2,618,426	created in step 7

TABLE II. LIST OF RECOVERED/CARVED PHOTOGRAPHS FROM THE SD CARD.

Filename	Source	Size	M time	A time	1st sector	Remark
?XYZ0001.JPG	recovered	508 KB	1/1/2008 9:34	6/1/2009	155	1st batch
?XYZ0002.JPG	recovered	0.5 MB	1/1/2008 9:35	6/1/2009	1243	1st batch
?XYZ0003.JPG	recovered	426 KB	1/1/2008 9:35	6/1/2009	2331	1st batch
?XYZ0004.JPG	recovered	402 KB	1/1/2008 9:36	6/1/2009	3195	1st batch
?XYZ0006.JPG	recovered	1.9 MB	9/1/2008 12:37	9/1/2008	3963	3rd batch
?XYZ0005.JPG	recovered	388 KB	1/1/2008 9:36	6/1/2009	4027	1st batch
Abcd123 2008-01-06 07:23:23.jpg	carved	1.3 MB	6/1/2008 7:23	unknown	9595	2nd batch
Abcd123 2008-01-06 06:48:55.jpg	carved	0.9 MB	6/1/2008 6:48	unknown	10555	2nd batch
?XYZ0049.JPG	recovered	319 KB	6/1/2008 11:36	5/11/2008	12315	2nd batch
?XYZ0050.JPG	recovered	264 KB	6/1/2008 11:33	5/11/2008	13115	2nd batch
Abcd123 2008-01-06 06:49:27.jpg	carved	1.0 MB	6/1/2008 6:49	unknown	18331	2nd batch
Abcd123 2008-01-06 07:07:15.jpg	carved	0.7 MB	6/1/2008 7:07	unknown	116731	2nd batch
Abcd123 2008-01-06 07:07:27.jpg	carved	0.6 MB	6/1/2008 7:07	unknown	118203	2nd batch
?XYZ0007.JPG	recovered	2.0 MB	6/1/2008 7:19	5/11/2008	120027	2nd batch
?XYZ0010.JPG	recovered	0.6 MB	6/1/2008 7:25	5/11/2008	174747	2nd batch
?XYZ0011.JPG	recovered	0.7 MB	6/1/2008 7:28	5/11/2008	175995	2nd batch
?XYZ0012.JPG	recovered	350 KB	6/1/2008 7:29	5/11/2008	239195	2nd batch
?XYZ0013.JPG	recovered	0.6 MB	6/1/2008 7:30	5/11/2008	239899	2nd batch
?XYZ0014.JPG	recovered	0.6 MB	6/1/2008 7:33	5/11/2008	241083	2nd batch
Abcd123 2008-01-06 07:34:34.jpg	carved	0.8 MB	6/1/2008 7:34	unknown	242363	2nd batch
?XYZ0016.JPG	recovered	0.9 MB	6/1/2008 7:36	5/11/2008	244123	2nd batch
?XYZ0017.JPG	recovered	353 KB	6/1/2008 7:37	5/11/2008	246043	2nd batch
?XYZ0018.JPG	recovered	227 KB	6/1/2008 7:38	5/11/2008	246779	2nd batch
?XYZ0019.JPG	recovered	1.4 MB	6/1/2008 7:46	5/11/2008	247259	2nd batch
?XYZ0020.JPG	recovered	0.9 MB	6/1/2008 7:48	5/11/2008	250235	2nd batch
?XYZ0021.JPG	recovered	0.6 MB	6/1/2008 7:49	5/11/2008	252123	2nd batch
?XYZ0022.JPG	recovered	409 KB	6/1/2008 7:50	5/11/2008	253307	2nd batch
?XYZ0023.JPG	recovered	0.7 MB	6/1/2008 7:52	5/11/2008	254139	2nd batch
?XYZ0024.JPG	recovered	1.4 MB	6/1/2008 7:52	5/11/2008	255675	2nd batch
?XYZ0029.JPG	recovered	0.6 MB	6/1/2008 7:57	5/11/2008	258619	2nd batch
?XYZ0026.JPG	recovered	0.7 MB	6/1/2008 7:54	5/11/2008	259867	2nd batch
?XYZ0030.JPG	recovered	0.7 MB	6/1/2008 8:00	5/11/2008	261275	2nd batch
?XYZ0028.JPG	recovered	0.5 MB	6/1/2008 7:55	5/11/2008	363067	2nd batch
?XYZ0031.JPG	recovered	0.6 MB	6/1/2008 8:00	5/11/2008	402715	2nd batch

Filename	Source	Size	M time	A time	1st sector	Remark
?XYZ0032.JPG	recovered	450 KB	6/1/2008 8:03	5/11/2008	404027	2nd batch
?XYZ0033.JPG	recovered	474 KB	6/1/2008 8:04	5/11/2008	404955	2nd batch
?XYZ0039.JPG	recovered	0.7 MB	6/1/2008 8:11	5/11/2008	405915	2nd batch
?XYZ0038.JPG	recovered	323 KB	6/1/2008 8:10	5/11/2008	406779	2nd batch
?XYZ0035.JPG	recovered	490 KB	6/1/2008 8:05	5/11/2008	407451	2nd batch
?XYZ0036.JPG	recovered	419 KB	6/1/2008 8:05	5/11/2008	408443	2nd batch
?XYZ0040.JPG	recovered	0.5 MB	6/1/2008 8:13	5/11/2008	409883	2nd batch
?XYZ0045.JPG	recovered	0.7 MB	6/1/2008 8:17	5/11/2008	410939	2nd batch
?XYZ0042.JPG	recovered	0.7 MB	6/1/2008 8:14	5/11/2008	418939	2nd batch
?XYZ0043.JPG	recovered	0.6 MB	6/1/2008 8:15	5/11/2008	420347	2nd batch
?XYZ0044.JPG	recovered	0.7 MB	6/1/2008 8:15	5/11/2008	421659	2nd batch
?XYZ0046.JPG	recovered	0.7 MB	6/1/2008 8:18	5/11/2008	423483	2nd batch
?XYZ0047.JPG	recovered	360 KB	6/1/2008 8:20	5/11/2008	424955	2nd batch
?XYZ0048.JPG	recovered	319 KB	6/1/2008 8:27	5/11/2008	425691	2nd batch
?XYZ0049.JPG	recovered	345 KB	6/1/2008 8:28	5/11/2008	426331	2nd batch
?XYZ0050.JPG	recovered	305 KB	6/1/2008 8:29	5/11/2008	427035	2nd batch
?XYZ0001.JPG	recovered	508 KB	1/1/2008 9:34	6/1/2009	155	2nd batch
?XYZ0002.JPG	recovered	0.5 MB	1/1/2008 9:35	6/1/2009	1243	2nd batch

REFERENCES

[1] K. Chow, F. Law, M. Kwan, P. Lai, "The Rules of Time on NTFS File System", 2nd International Workshop on Systematic Approaches to Digital Forensic Engineering, Seattle, Washington, April 2007

[2] S. Willassen, "Finding Evidence of Antedating in Digital Investigations", ARES 2008, Barcelona, Spain, March 2008

[3] SD Association, "SD Technology", Available at: http://www.sdcard.org/developers/tech/

[4] Microsoft, "FAT32 File System Specification", Available at: http://www.microsoft.com/taiwan/whdc/system/platform/firmware/fatgen.mspx

[5] Microsoft, "How NTFS works", Available at: http://technet.microsoft.com/en-us/library/cc781134%28WS.10%29.aspx

[6] JEITA, "Design rule for Camera File system DCF Version 2.0 (English version)", Available at: http://www.jeita.or.jp/cgi%2Dbin/standard_e/pdfpage.cgi?jk_n=51

[7] JEITA, "Exchangeable image file format for digital still cameras: Exif Version 2.2 (English version)", Available at: http://www.jeita.or.jp/cgi%2Dbin/standard_e/pdfpage.cgi?jk_n=47

[8] Microsoft Knowledge Base, "Time Stamps Change When Copying From NTFS to FAT", Available at: http://support.microsoft.com/kb/127830

[9] Bryce Cogswell, "DiskView", Available at http://technet.microsoft.com/en%2Dus/sysinternals/bb896650.aspx