Identification of Fragmented JPEG Files in the Absence of File Systems

Nur Fasihah Abdul Kadir
Faculty of Computing, Universiti Tecknologi Malaysia Skudai 81310, Malaysia
nfasihah5@live.utm.my

Shukor Abd Razak
Faculty of Computing, Universiti Teknologi Malaysia Skudai 81310, Malaysia
shukorar@utm.my

Hassan Chizari
Faculty of Computing, Universiti Teknologi Malaysia Skudai 81310, Malaysia
chizari@utm.my

***Abstract*--- Identifying fragmented and deleted files from scattered digital storage become crucial needs in computer forensic. Storage media experience regular space fragmentation which gives direct consequence to the files system series. This paper specifies a case where the jpeg files are heavily fragmented with absent file header which contains maximum information for the stored data can be easily retrieved. The problem is formulated using statistical byte frequency analysis for identifying the group of jpeg file fragments. Several related works have addressed the issue of classifying variety types of file format with high occurrence of being fragmented such as avi, doc, wav file and etc. These files have been tagged as among the larger file format. We provide techniques for identifying the pattern of file fragments distribution and describe roles of selected clustering attributes. Finally, we provide experimental results presenting that the jpeg fragments distribution can be retrieved with quite small gap differences between the groups.**

Keywords*--- *byte frequency; JPEG file format; image processing; file carving; file identification; statistical methods

I. INTRODUCTION

A field of computer science determine image as an exact replica of an object, which stored inside the storage device. Computer storage represent image in a digital format. Digital images play a vital role to the resolution of advertising, education and filming activities. In business world, image roles as an instant communication to present products and services prompt to the market. It shows that images are very useful in most industries. With the high density of image processing technology, make image more interactive to be modified, to comply the certain preferences. However, this kind of adjustment will disturb the originality of raw data.

The raw image can be recognized with minimum processed data on it which is produced by optical device such as digital camera. The raw data is significance in digital investigation process in order to fight for justice towards any criminal activities. In fact, data organization inside storage also reflects the quality of image itself. Image experience "compress" and "decompress" of bit string to save the storage space. Furthermore, it performs data segmentation on chunk space of the memory. The data segmentation phenomenon extends challenge to researchers on identifying these segmented image files using several potential techniques or signatures.

In this study, further analysis will take place on JPEG file format. JPEG files image is also known as JFIF format. JPEG is a standard image encoding besides GIF, BMP, and PNG etc. It is used widely in digital storage equipment [1]. The JPEG file format always comes with maximum quality and possesses a larger file size. However, for some purposes such as email, web pages and memory cards, they require small file size. Most of the JPEG files are designed with lossy compression features by purposely eliminate certain percentage of original data to address those demands. In this case, the encoded bytes distribution which belongs to each JPEG file should be different from others due to its unique encoding pattern. This study highlights the nature of JPEG files and proposes alternative solution to file recovery technique in order to address the files deleted and segmented issue.

A. Motivation

Physical forensic evidences are not as simple to be handled like common testimonial-based evidence which relay on spoken work or statements. An image file like JPEG is one type of physical evidence in digital format. The digital evidences are less sturdiness, which easily can be tampered, sometimes remains subtle without sharp changes. This type of evidence is also hard to tolerate with any degree of carelessness when conducting identification or collation processes.

JPEG file could be found at anywhere such like cellular phone and computer devices, their usage is almost universal. Those devices are nowadays worked on-line with high density of computation and scalable storage, for information dissemination. The JPEG files are made up of convenient to advanced optical devices, by default. Hence, it grabbed an interest for editing tool to work on them. It also occupied with compression features that can reduce storage space and power load.

JPEG files use EXIF scheme for its metadata. The raw data that is embedded inside file system metadata is always a source spot for researcher to craft the content of image files. The content inside the file header is worthful to identify JPEG segment, but it is undependable if in some cases the header tag is missing.

B. Problem Statement

File carving analysis takes a huge portion in digital investigation. From last few decades, several existing techniques have been tried to deal with deleted files from a hard disk compartment. However, most of them least noted the biggest issues in data carving which somehow deleted files were embedded in a form of segments. Even though, if they are afforded to render solution for that problem, even then, the carving procedures still depend on file system metadata which does not guarantee to be always available (damaged or corrupted). The researchers need to realize that the damaged or deleted data might be due to intentional behaviour like criminal activities.

II. Related Work

In the court of law, the maximum and authentic data is crucial to deal with most of the digital cases. Therefore, it comes mandatory to provide those data by retrieving various types of data such as hidden, deleted, compressed, encrypted and also segmented data that embedded within storage devices. Besides that, the issue of accuracy and efficiency of carver tool should always be emphasized, so then, significant data could be revealed in any particular time, and be ranked based on its quality level. EXCAVATOR system is one of example to measure runtime performance of carving JPEG files. It is evaluated with other tools like Revivelt, Scalpel and Photorec. From the evaluation, it becomes the most efficient method but remain the same accuracy with Revivelt. After made some enhancement, [2]suggest this tool to be used by forensic investigator without make changes on its architecture. The better performance and accuracy is crucially needed by carving tools in order to provide authentic data and present meaningful result in court of law.

A. File System Metadata

In previous works, the researchers still were depending on file header [3] to retrieve other information like image resolution, the entire size of an image. Then, from resolution information, they could carve the width of the image, and thus the amount of usable pixels is certainly can be recognized. The same condition is pointed to the footer of the files as well. These are the studies that have used this approach[4];[5];[6]; and[7].

In other cases, the object validation techniques have been used to emphasize more on bits sequence of one segment to other possible segments. There are three types of parameter used in [4]. It consists of validating headers and footers; validating container structure which is comprised of file system metadata, coloration table and Huffman encoded codes; validating actual data (decompression); semantic validation which is based on human language (related to written documents); and manual validation (naked-eyes examination). Their attempt is to work for identification and reassembling file types. Garfilkel was deal

with drives that not equally segmented. He utilized *Bisegment Gap Carving* algorithm to demonstrate the carving process. This technique examined every single file object and attempted to recover all pieces of specific files, however, it experienced false-positive on the result.

Reference [5] emphasized on identification, reassembling and segmentation point detection of file carving procedures. This work meant to cover some enhancement in [3] and [4] which used dataset from DFRWS (containing JPEG files, variety of Microsoft Office files and ZIP files. Based on the findings, they had difficulty to identify all segmented points. By default, the validator will discards unresolved blocks if false addition could not be avoided. It seems like segmented files are not such a simple matters to deal with.

Other important piece of work belongs to [6] who proposed mapping function of determining errors in decoded image files. The discriminator is used to validate, whether the carved file is corrupt or not. The edge detection technique is also used to minimize its error rate. He used *libjpeg* to decode the JPEG files. It operates by identifying the discontinuous sector offset as accurate as possible. This tool has ability to detect error occurrence in early stage of decoding process. However sometimes it could not manage the errors within the compressed bit streams. Cohen tried to propose solution for DFRWS challenges. He also raised a point that DFRWS datasets are actually not conform to the real file system instead are designed to generate extreme scenarios with diversity form of files system. This paper is worked for identification and reassembling files.

In addition, [8] came with *myKarve* to address the limitation in earlier file carving tools like *Scalpe* and *Foremost* which was less concerned about segmented files. So, their work was based on point detection of segmented files as well as the paper from[9].

As mentioned in [10] and [7], they were using Huffman code table also known as Define Huffman Table (DHT) which matching the bits sequence to recover the next file segments. Reference [7] used additional parameters like MCUs, AC and DC values and Huffman codes. These parameters have potential to solve the issue of identification, point detection and reassembling of segmented or deleted files. They highlighted, restart marker is also a kind of item that could be utilized to resolve bit stream errors. However, in the case of all image are encoded into same set of Huffman table, then their approach is no longer beneficial. Moreover, the restart markers is an optional signature, as it mostly generated for large size images.

B. Statistical Approach

Reference [11] determined a method using average, distribution of averages, standard deviation, distribution of standard deviation and kurtosis, were adequate to effectively give class to different group of data types. Based on supportive literature and analysis, they divided the class into two cases, 256-byte and 1024-byte. The method is slightly supported by plotting the behavior of these statistic measurements over each specific file. In 2008, [12] utilized byte value distribution as additional measures to the previous work. They expend the method in quiet detail and provide a secondary analysis to give clear picture of overall findings.

Reference [13] adopted byte frequency distribution with Shannon entropy and Kolmogorov to his approach. He chunked the bytes into 4096-byte which equivalent to hardisk cluster size. Eleven different file types were analyzed. He displayed the obtained result in confusion matrix to recognize the effect of false-positive and program failure. Almost all the html and jpeg files are recognized with 99% and 98% respectively, while others show ambiguous result between 18% to 78% recognition rates. Follow-up work in [14] attempt extra measures (16 statistical attributes) upon the classification program. Subtle distinction took place for three compressed format: jpeg, pdf and gif. He excluded unreliable file system metadata for the analysis.

Reference [15] presented deflate classifier by purpose to differentiate compressed data from encrypted data. Deflate is tree-classifier program designed to identify from the most generic to more specific data class. The classifier rely also on file header and sustainable for unaddressed fragments when header is not available. He adopted Shannon's formula over data entropy and ASCII codes to well-defined encoding format.

III. REDEFINE THE PROBLEM

The prior work in the area of file fragment classification could be summarized into two critical issues. First, most of the existing work unable to address one of the biggest issues in data recovery or data carving which is how to recover deleted files if the files was stored in the segmented form in a huge size of hard disk. Second, even if there are works that try to provide solutions to the problem mentioned, most of them have made an assumption that the file system was still intact. Such an assumption is not always true because in some cases the file system is damaged or intentionally deleted to erase all criminal traces. Below, described the root cause of file storage pattern.

Fig. 1 illustrates several cases of file system within data storage. The figure shows that File *A* has been split into four segments consists of four blocks and it contains header and footer markers; file *C* has been broken into three segments consists of six blocks and without header marker; file *F* comes with missing footer marker; while file *E* is stored in complete structure and in correct order. All those segments are distributed across 30 logical blocks.

Regularly, files are created, modified and deleted on the storage so most file systems are forced to be segmented. The segmentation occurs when the files store across the limit of cluster size and in most cases it rarely constructs in the correct sequence. In current works, the researchers are still struggling to find the solution on how to recover deleted files from a hard disk which exist in segmented form. As a result, the existing carving tools can only recover partial or sometimes incorrect reconstruction of segmented image [3].

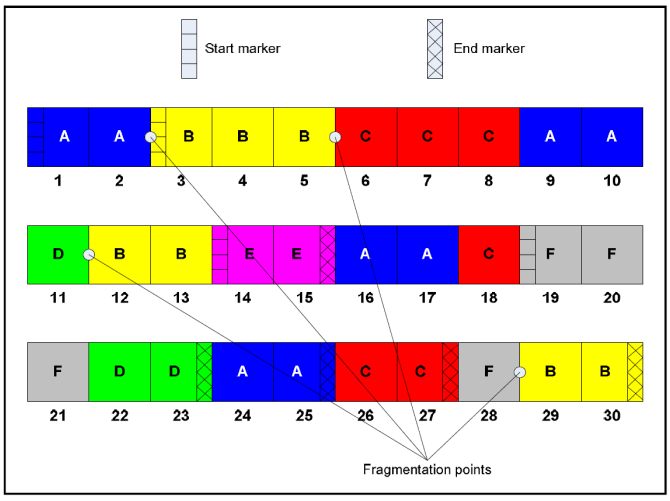

Fig. 1. Concept of file storage pattern

IV. DESIGN SCHEME

There are four different JPEG files used in this work, which are selected based on their reputation as among the larger file format, thus more possible to become segmented. JPEG files format are highly significant to the forensic investigators and get extra attention from researchers to address the issue of likely being segmented [4]. In average, about 285 segments of 512 bytes are generated from each JPEG file. Specifically, 341 segments taken from *file1*, 306 segments taken from *file2*, 250 segments taken from *file3* and 264 segments taken from *file4*. These files were collected from Caltech (California Institute of Technology) Image Database by Markus Weber. All files are taken from frontal face dataset which made up from same dimensional of 986 x 592 pixels with different backgrounds, lighting and face expressions. These files were then downloaded and added to dataset.

A. *Segmentation Program*

The file information is sieved and chunked into length of 512-byte per segment. The process is significant to meet the nature of file storage that stored information in minimum 512-byte of sector size. Cut off files data into 512-byte length could address the deleted file issue while 4096-byte length could cover the issue of segmented files. The file segmentation program is as follows (refer Figure 2).

```
% Open .bin file;
% Read .bin file in 512-byte length;
% Transpose byte sequence from row into column;
% End files reading;
```

Fig. 2. Code comments of 512-byte segmentation program

The training file is first viewed in hexadecimal editor tool. By using the hex tool, the file segments are then sieved and saved as .bin file to preserve the originality of file content. Then, Matlab Application is used to generate matrix from created .bin file. The .bin file is then read into byte stream buffer storage of base-10 numerical value. In order to meet specification of hard disk storage, the formed matrix is organized into length of 512 bytes per row which equal to the size of storage sector. The initial and the end frame of files are consisting of markers which indicate file format. Then these sections need to be removed from the training set.

B. *Statistical Byte Frequency Program*

The statistical byte frequency (SBF) is a program that provide alternative for file carving system when file signatures are disable dues to deleted or corrupted file system metadata. In this work, the SBF is expected to retrieve some pattern by implementing its statistic parameter to differentiate a unique pattern among different JPEG files. File pattern is contingent upon its bytes/pixels distribution, and different bytes distribution may give a clue that the files are made up from different JPEG file. To determine a unique file pattern independently is not an easy job. It should involve classifier (attributes) to create some regions whereby objects (files) can be distinguished based on its characteristics.

For the purpose of this research, there are 17 attributes are chose in the program. They are selected based on their tendencies to group objects. More specifically mean median and mode attributes, geometric mean, harmean, high and low frequency, ASCII frequency, standard deviation, variance, standard deviation frequency, index of coincidence, mean absolute deviation, entropy, chi-square, kolmogorov and hamming weight attributes are well involved in classification and clustering mechanism.

This work implements the methodology of bootstrap which relied under data resampling group. Data resampling works in the way that some items are selected at random from a population and used to validate the hypotheses about the population. The bootstrap maintains unsupervised operation towards the experimental data. Bootstrapping procedures have ability to adjust the standard errors of bytes distribution, and in fact this is important part involved in any statistical analysis. Bootstrap gives significant in data clustering by drawing the cluster units with replacement instead of drawing the observation units with replacement. For instance, there are 300 segments from 4 different JPEG files. The non-clustered bootstrap draws 300 observations from those 300 unit segments with replacement for each repetition. While the clustered bootstrap will rather draws 4 unit of JPEG files format with replacement. Furthermore, this study manage to provide numbers of round of bootstrapping process till the pattern of file segments is obtained. After went through several trial, three rounds is decided to be implemented in testing phase. In this experiment, the three round of bootstrapping selection is thought to be the better approach in finding four different JPEG file segments. Figure 3 showed the work flow of statistical byte frequency program in two stages.

First Stage (Calculate data segment based on selected attributes)

```
% Input .txt file content into the program;
% Apply calculation to each of the 512-byte segment based on those 17 selected attributes;
% Export calculated data into an array;
```

Second Stage (Export attributes into bootstrap iteration)

```
% Read the attribute arrays from previous stage;
% Resample data into bootstrap function;
(iterate this process until it recognizes some unique pattern;
```

Fig. 3. Code comments for statistical byte frequency

This study has described thoroughly the overall process of data preprocessing which involved huge number of dataset. The collected data is used for statistical byte frequency analysis and its results are finally carried into classification program to visualize the relationship between attributes in a clear manner. The k-mean clustering was elected as classifier program since it is a well-known and work very well in large dataset. The model design is utilized to solve the issue of identifying file type inside hard disk storage in segmented state without using file system metadata.

The results of the statistical byte frequency technique in term of accuracy and performance speed briefly discussed. The testing program was utilized using four different JPEG files format. More specific, the minimum, maximum and average values of byte distribution based on the clustering attributes are taking into account. The rational to use minimum and maximum values are to ensure no overlapping occur between the segments of the different JPEG files, thus provide a division range to distinguish the groups of segments. While, the average value is actually represents the overall reading of each file group.

V. RESULT AND ANALYSIS

As mentioned earlier, byte frequency analysis used the bootstrap function to find pattern of byte distribution based from 17 selected statistic attributes. By using this function, researcher could identify certain pattern based on attribute values after went through exhaustive process of multiple round iteration. It computes statistics on each data sample using *@mean* function variable, and returns the computed result in the matrix *tJPG* which example for JPEG file sample. Table 1 until Table 3 below shows the final reading of bootstrap iteration from four different JPEG files based on its minimum, maximum and average values. The results are presented in sort of table together with a brief summary of the findings. See how the characters of each attribute play their roll in retrieving group of byte distribution.

Table 1 indicates quite a clear gap of mean, median and mode distribution based on its minimum and maximum value between all JPEG files. The mean, median and mode are the simplest statistical formula to calculate and the easiest to understand for clustering analysis. In overall, closer reading were obtained from *file1* and *file2* with only 0.057 differences, 0.406 difference and 0.363 differences of mean, median and mode values, respectively. Based on these three attributes, mode attribute has potential to draw a clearest gap to differentiate segments of JPEG files with the averages of 84.393 from *file1*, 84.756 from *file2*, 87.055 from *file3* and 71.371 from *file4*. The rating is followed by median and mean, respectively. Additionally, *file3* showed quite a huge dissimilarity of byte distribution compared to all based on these three class attributes.

TABLE 1 THE MINIMUM, MAXIMUM AND AVERAGE VALUE OF ATTRIBUTES MEAN, MEDIAN AND MODE

JPEG file	Mean		Median		Mode	
	min.	max.	min.	max.	min.	max.
file1	129.385	129.397	129.982	129.998	84.356	84.341
	average		average		average	
	129.391		129.989		84.393	
	min.	max.	min.	max.	min.	max.
file2	129.329	129.339	129.575	129.589	84.712	84.802
	average		average		average	
	129.334		129.583		84.756	
	min.	max.	min.	max.	min.	max.
file3	132.146	132.160	132.850	132.880	87.990	87.116
	average		average		average	
	132.153		132.866		87.055	
	min.	max.	min.	max.	min.	max.
file4	129.447	129.466	129.866	129.886	71.371	71.424
	average		average		average	
	129.456		129.875		71.371	

Entropy attribute is a function that well implemented for numeric data in extremely large dataset. It this work, it measured the amount of disorder in the byte distribution. Entropy function holds the value of 0.0 for its smallest possible value when all characters of the byte are same. Table 2 shows the *file3* consists of least variation of byte value compared to three other JPEG files. While, the *file1* file indicates the highest variation of byte character among all. For Kolmogorov attribute, it gives probability of complex structure of byte stream. Shown that *file1* gives highest value of Kolmogorov which made up from complex density of pixel distribution compared to others. Chi-square function works actual numbers instead of proportions, means, percentages, etc. As byte distribution consists of solid actual number, so chi square function could be utilized to find out its pattern. The small probability of occurrence (e.g.: $p<0.05$) indicates to high connection exist between byte distribution. The smaller occurrence probability, led to larger value of chi-square. In this case, the *file3* shows highest connection within its byte distribution, thus proves that *file3* consists of associated byte values.

TABLE 2 THE MINIMUM, MAXIMUM AND AVERAGE VALUE OF ATTRIBUTES ENTROPY, KOLMOGOROV AND CHI-SQUARE

JPEG file	Entropy		Kolmogorov		Chi-square	
	min.	max.	min.	max.	min.	max.
file1	7.476	7.476	5.778	5.778	382.539	382.546
	average		average		average	
	7.476		5.778		382.542	
	min.	max.	min.	max.	min.	max.
file2	7.457	7.458	5.766	5.766	382.887	382.895
	average		average		average	
	7.457		5.766		382.891	
	min.	max.	min.	max.	min.	max.
file3	7.363	7.363	5.680	5.681	386.400	386.418
	average		average		average	
	7.363		5.680		386.408	
	min.	max.	min.	max.	min.	max.
file4	7.390	7.391	5.701	5.702	385.832	385.846
	average		average		average	
	7.391		5.702		385.838	

Hamming weight function estimates number of nonzero components that belong to a string. As shown in Table 3, hamming weight of *file3* is among the largest compared to other four JPEG files. It indicates that the *file3* contained higher number of nonzero components on its 8-bit strings. However, the different to *file4* file is only 0.003, mean such a close amount. In case of mean absolute deviation, *file4* shows the highest result followed by *file3* file, with 0.032 different averagely. This function calculates value of mean in twice. One is the mean of each segment; another is the distance mean between two segments.

TABLE 3 THE MINIMUM, MAXIMUM AND AVERAGE VALUE OF ATTRIBUTES HAMMING WEIGHT AND MEAN ABSOLUTE DEVIATION

JPEG file	Hamming weight		Mean absolute deviation	
	min.	max.	min.	max.
file1	0.504	0.504	64.777	64.779
	average		average	
	0.504		64.778	
	min.	max.	min.	max.
file2	0.516	0.517	64.954	64.957
	average		average	
	0.517		64.956	
	min.	max.	min.	max.
file3	0.525	0.526	66.141	66.144
	average		average	
	0.525		66.142	
	min.	max.	min.	max.
file4	0.521	0.522	66.172	66.176
	average		average	
	0.522		66.174	

VI. DISCUSSION

In order to identify the groups of segment with averagely 290 segments per file, it took about 2 to 5 seconds to accomplish the task. As mentioned in previous chapter, the testing process took three rounds bootstrapping in order to retrieve the pattern of file segments. The additional and reduction number of round could surely effect the overall time taken to identify the byte pattern of four different JPEG files. Besides that, the numbers of segments involved also give direct consequence to computational speed and thus give a benchmark whether the technique is convenient for big amount of dataset or vice versa.

As the conclusion, researcher validated the model of statistical byte frequency analysis in term of 17 clustering attributes which actually indicates the accuracy of implemented technique. Based from the result, not all attributes could provide a good classifier. Moreover, the excessive number of round of bootstrap iteration could eliminate the variation of byte distribution which then leads to misclassified file segments. While, small number of round could deviates the distribution of bytes, thus hard to retrieve the pattern between different JPEG files.

VII. CONCLUSION

This research focused on the statistical byte frequency analysis where the JPEG file segments can be identified without depending to file system metadata which might sometimes corrupt. The analysis takes into account the nature of segmented hard disk memory and address the issue of deleted segments. The science behind byte frequency, give an insight that files are made up from several unique patterns which could be retrieved using mathematical approach (statistic).It is concluded that the model design is convenient to be implemented for real conduct of file carving program. The file carving program is significantly needed to deal with various criminal cases of hard disk drive. The study is helpful to identify JPEG file segments in segmented memory storage with take into account the analysis of byte frequencies.

Like other work, the proposed solution also comes with limitations. The limitations of the study are: the proposed solution is evaluated with only four different JPEG files. For a better result, extra number of JPEG files and segments are recommended to be evaluated; the inappropriate number of bootstrap iteration might hide the pattern of byte distribution.

VIII. FUTURE WORKS

File carving program has been worked for a decade, but have limitations with its approach. Therefore, the future researchers are invited to be apart, and get hands dirty with this field. The program response time might become one of the issues in identifying pattern of byte streams inside storage device once it deal with large scale of dataset. Large dataset always gives undesirable qualities to efficiency of the carving program. In future works, the program is targeted to provide high accuracy, excellent time performance and applicable to all operating system. Besides, the file program need consider if the dataset contains extra number of JPEG files. It could be the additional challenge to the researchers to differentiate between huge numbers of JPEG files in storage device.

REFERENCES

[1] J. van den Bos and T. van der Storm, "Bringing domain-specific languages to digital forensics," presented at the Software Engineering (ICSE), 2011.

[2] J. van den Bos and T. van der Storm, "Domain-specific optimization in digital forensics," in Theory and Practice of Model Transformations, ed Prague: Springer Berlin Heidelberg, 2012, pp. 121-136.

[3] N. Memon and A. Pal, "Automated reassembly of file fragmented images using greedy algorithms," *Image Processing,* vol. 15, pp. 385-393, 2006.

[4] S. L. Garfinkel, "Carving contiguous and fragmented files with fast object validation," *Digital Investigation,* vol. 4, pp. 2-12, 2007.

[5] A. Pal, H. T. Sencar, and N. Memon, "Detecting file fragmentation point using sequential hypothesis testing," *Digital Investigation,* vol. 5, Supplement, pp. 2-13, 2008.

[6] M. I. Cohen, "Advanced JPEG Carving," ed, 2008.

[7] H. T. Sencar and N. Memon, "Identification and recovery of JPEG files with missing fragments," *Digital Investigation,* vol. 6, Supplement, pp. 88-98, 2009.

[8] K. M. Mohamad, A. Patel, and M. M. Deris, "Carving JPEG images and thumbnails using image pattern matching," in *Computers & Informatics (ISCI)*, 2011, pp. 78-83.

[9] L. Qiming, B. Sahin, E. C. Chang, and V. L. L. Thing, "Content based JPEG fragmentation point detection," presented at the Multimedia and Expo (ICME), 2011.

[10] K. M. Mohamad and M. M. Deris, "Fragmentation Point Detection of JPEG Images at DHT Using Validator," in *Future Generation Information Technology*, ed: Springer Berlin Heidelberg, 2009, pp. 173-180.

[11] R. F. Erbacher and J. Mulholland, "Identification and Localization of Data Types within Large-Scale File Systems," in *Systematic Approaches to Digital Forensic Engineering*, Bell Harbor, WA, 2007, pp. 55-70.

[12] S. J. Moody and R. F. Erbacher, "SADI - Statistical Analysis for Data Type Identification," in *Systematic Approaches to Digital Forensic Engineering*, Oakland, CA, 2008, pp. 41-54.

[13] C. J. Veenman, "Statistical Disk Cluster Classification for File Carving," in *Information Assurance and Security, 2007*, Manchester, 2007, pp. 393-398.

[14] W. C. Calhoun and D. Coles, "Predicting the types of file fragments," *Digital Investigation,* vol. 5, Supplement, pp. 14-20, 2008.

[15] V. Roussev and C. Quates, "File fragment encoding classification—An empirical approach," *Digital Investigation,* vol. 10, Supplement, pp. 69-77, 2013.