Reassembling the Fragmented JPEG Images Based on Sequential Pixel Prediction

MingXu
Institute of Computer Application Technology, Hangzhou Dianzi University
Hangzhou 310018, P. R. China
mxu@hdu.edu.cn

Shule Dong
Institute of Computer Application Technology, Hangzhou Dianzi University
Hangzhou 310018, P. R. China
leon.hziee@gmail.com

Abstract—**File carving is a digital forensic technique which data files are extracted from unstructured data sources without the information of the previously stored file system. Since JPEG is one of the most popular image formats in the storage and distribution of digital photographic imagery, JPEG Carving aroused numerous researchers' great interest. In this paper, a new algorithm of reassembling the fragmented JPEG file is described. To reassemble the fragmented JPEG file, a sequential pixel prediction model based on the neural network was proposed to determine whether or not the given fragment is adjacent to the last fragment in the original. The performance results obtained from the fragmented test-sets of DFRWS 2006 show that the method can be effectively used in recovery of fragmented JPEG files.**

Keywords-component; data recovery; file carving; BP algorithm; digital forensics

I. INTRODUCTION

Nowadays, with this huge increase in digital data storage, the need to recover data due to human error, device malfunction, computer criminal activities or deliberate sabotage has also increased. For example a criminal fragments a digital image into small pieces hidden in other digital data, so the police can't recreate the original image because the file is already fragmented. Consequently there is a need for new technology that is able to work with fragmented data held on the unstructured original disk image. File carving is a forensics technique that recovers files based merely on file structure and content and without any matching file system meta-data. File carving is most often used to recover files from damaged or incomplete file systems and unallocated space in a drive. [1]

The usefulness of file carving technology has aroused numerous researchers' great interest. Most famous research organization is the Digital Forensic Research Work-Shop (DFRWS). They raised forensic challenge every year, and these challenges have posed extremely difficult images, encouraging the researchers to develop the techniques and tools to advance the state of the art. [2]

The paper focuses on the JPEG carving problem, assuming that the only information present is the actual contents of the fragments and any file table records indicating a fragment's link to an image file is unavailable. This paper proposes a new approach, which using neural network model to predict the pixels of next fragment for JPEG carving.

The next section describes the related work of the JPEG file carving. Section 3 begins with a description of fragmentation and how to carve the fragment JPEG file, and then we describe three simple linear predictive techniques. Section 4 describes our technique for using BP algorithm to carve fragmented JPEG file. We detail our experiments and results in section 5. At last, we conclude in section 6 with a discussion on future work.

II. RELATED WORK

The initial type of carvers was simple Start of File/End of File (SOF/EOF) carvers. These simply analyze headers and footers of a file and attempt to merge all the blocks in between. One of the most well known of these carvers is Scalpel [3]. Garfinkel [4] presents a method called Bifragment Gap Carving (BGC), which can reconnect a file fragmented into two fragments separated by one or more junk hard disk sectors. But the drawback of the algorithm is that it can only deal with files fragmented into two parts.

Cohen [5, 6] presents a novel theory describing carving as a mathematical construction of a mapping function between the file bytes and the image bytes. In the paper, Cohen describes two semantic carvers for PDF, Zip and JPEG files. The carvers utilize the internal structure of the three file types to identify and reassemble fragments. In the JPEG file carving, he uses an edge detection technique to estimate the error levels. The method Cohen uses only information related to the headers and internal structure and the edge detection technique can not deal with all the fragments JPEG files.

Memon and his research group have research how to reassemble fragmented bitmap images for years.[7, 8, 9, 10] They describe the image reassemble problem as K-vertex disjoint graph problem and use different path optimizing algorithms to find the correct fragment pair sequence. To do this they have used different ways to assign weight to each possible fragment pair. But they assume each fragment

This work is supported by the Natural Science Foundation of Zhejiang Province (No. Y1090114) , and the Science and Technology Program of Zhejiang Province (No: 2008C21075).

978-1-4244-5273-6/09/$26.00 ©2009 IEEE

contains at least an image width of data which realities can not be.

Pal, student of Memon, further improve the performance of Garfunkel's bifragment gap carving technique and Pal and Memon's Parallel Unique Path technique for recovering fragmented files. In Pal's sequential fragmentation point detection method [11], the weights are evaluated sequentially and the test is ended in favor of a decision only when the resulting decision statistic is significantly low or high.

Martin Karrensand [12] presents a method to reassemble fragmented JPEG images containing Restart markers. They decode the luminance DC values in all restart intervals to form DC value chains, and identify the distance to its closest match using a sliding windows approach for the DC value chains. When two fragments are correctly connected vertically oriented lines are repeated in the DC value chain at an interval equaling of the image width. But the method currently only handles JPEG images containing RST markers.

Our image fragment reassembly method is similar to A. Pal and N. Memon's method. They used a simple linear predictive technique to assign weights to fragment pairs, but we use Error Back Propagation (BP) predictive technique to reassemble fragment JPEG file. Moreover, they also require each fragment contains at least an image width of data, but our method does not need this condition.

III. CARVING METHOD FOR FRAGMENTED JPEG

A. *Fragment Problem*

File fragmentation is an unavoidable problem that affects many computers using a variety of file systems. Low disk space, appending/editing files or deliberate sabotage can cause the fragmentation. Fig 1 is a very simple example of a disk with 7 clusters to explain how fragment occurs.

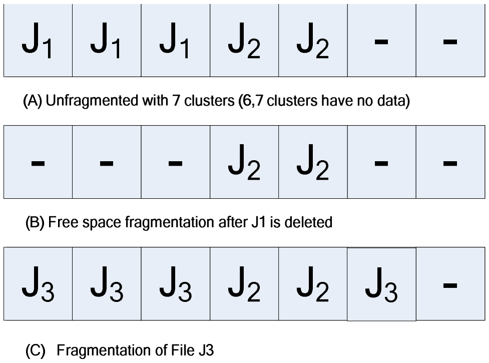

J1 J1 J1 J2 J2 - -
(A) Unfragmented with 7 clusters (6,7 clusters have no data)
- - - J2 J2 - -
(B) Free space fragmentation after J1 is deleted
J3 J3 J3 J2 J2 J3 -
(C) Fragmentation of File J3

Figure 1 A simplified example of file fragmentation

In the figure, the J_3 has been broken into two fragments. The first fragment starts at block 1 to block 3 and the second fragment starts at block 6.

B. *Fragment JPEG Carving*

To recover fragmented JPEG files correctly a file carver method must be able to determine the starting point of the JPEG file FFD8 and correct blocks that are required to reconstruct the file. It usually follows this three step process:

1) *Identify starting point of a JPEG file.*
2) *Identify blocks belonging to the file.*
3) *Order the blocks correctly to reconstruct the file.*

The main focus of this paper is to identify blocks correctly belonging to the file. We assumed a more realistic fragmentation scenario where fragments are not randomly scattered but have multiple blocks sequentially stored.

C. *Identify blocks*

In Fig 1, how can identify block 4, 5 not belong to the file J_3 is very important to the JPEG carving. It can be described that there is a set $\{A_0, A_1..., A_n\}$ of fragments of an image A, Suppose that A_i is one of fragments of the original image A, how can we identify whether fragment A_j belongs to the original image A or not. One simple technique to do this, is to prove fragment pairs A_i and A_j that are adjacent in the original image A.

It is also known that the JPEG image consists mostly of smooth regions and the edges present have a structure that can often be captured by simple linear predictive techniques. Hence another way to assess the likelihood that two image fragments A_j, A_i are indeed adjacent in the original image A is to compute prediction errors based on some simple linear predictive techniques. That is, prediction errors are computed for pixels in the last row of the front fragment A_i and the pixels in the first row of the back fragment A_j. Fig. 2

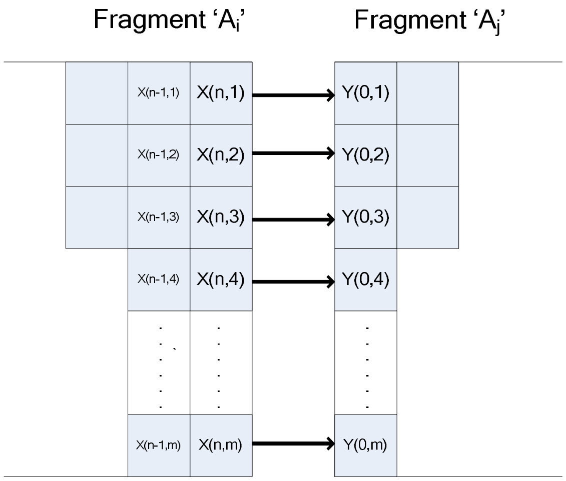


Figure 2 Use pixels in the last row of the front fragment A_i to predict the pixels in the first row of the back fragment A_j

There are some techniques that can be used to compute the likelihood based on gradient analysis across the boundaries of each pair of fragments. These techniques are mentioned by Memon[8].

- a Pixel Matching (PM): This is the simplest technique whereby the total number of pixels matching along the edges of size for the two fragments are summed. PM will compare each numbered pixel in fragment A_i matched in the value with the same numbered pixel in fragment A_j.The technique simple predict the value of the pixels in the first row of the back fragment

$$\hat{Y}(0,m) = X(n,m) \text{ .(1)}$$

- Sum Of Differences (SoD): The sum of differences is calculated across the RGB pixel values of the edge for the two fragments. SoD predict the value of the pixels in the first row of the back fragment

$$\left|\hat{Y}(0,m) - X(n.m)\right| = \left|X(n,m) - X(n-1,m)\right| \text{ .(2)}$$

- Median Edge Detection (MED): MED describe that each pixel is predicted from the value of the pixel above, to the left and left diagonal to it. So the value of the pixels in the first row of the back fragment

$$\hat{Y}(0,m) = \begin{cases} \min(X(n,m), Y(0,m-1)), X(n-1,m-1) \geq \max(X(n,m), Y(0,m-1)) & (3) \\ \max(X(n,m), Y(0,m-1)), X(n-1,m-1) \leq \min(X(n,m), Y(0,m-1)) \\ X(n,m) + Y(0,m-1) - X(n-1,m-1), \mathit{otherwise} \end{cases}$$

Where n is the fragments' width , m is the fragments' height.

Experimentally, we found MED techniques more useful than the PM and SoD techniques, but still not good enough. In the following section, the paper describes a new technique to predict the value of the pixels in the first row of the back fragment.

IV. Pixel Prediction using Error Back Propagation (BP) algorithm

This section proposes a pixel prediction method which is then utilized as a part of a JPEG image file recovery method.

A. *Problem formulation*

Suppose the JPEG file A has a orderly set $\{A_0, A_1, \ldots, A_n\}$ of fragments. Our objective is to determine the order in which fragments A_j need to be concatenated to the A_i ($i \in [0,1,\ldots,n$-1]. We assume that fragments are not randomly scattered but have multiple blocks sequentially stored. For example, the block number j of fragment A_j is greater than the block number i of fragment A_i in the disk image.(j>i).

Note that in order to determine the correct fragment reordering, we need to identify fragment pairs that are adjacent in the original file. One technique to do this is to compute the likelihood that fragment A_j follows A_i.

It is already known that the value of the pixels both fragments, and it is also known that the JPEG image consists mostly of smooth regions and the edges present have a structure that can often be captured by simple linear predictive techniques. The problem of how to compute the likelihood can be formulated as a pixel prediction problem. We use the value of the pixels $x(i,j)$ which belong to the front fragment A_i to predict the next fragment's pixels value $\hat{y}(i,j)$, and then compare between the predict value $\hat{y}(i,j)$ and the pixels value $y(i,j)$ that belong to the given data fragment A_j.

$$\hat{y}(i,j) = f(x(i,j)).\text{(4)}$$

B. *Back-Propagation (BP) Neural Networks*

In order to understand BP Neural Networks more clearly, we'll discuss a generalized network first.

1) The Feed-Forward Neural Network Model

Referring to Fig.3, the network function is following: Each neuron receives a signal from the neurons in the previous layer, and each of those signals is multiplied by a separate weight value. The weighted inputs are summed, and passed through a limiting function which scales the output to a fixed range of values. The output of the limiter is then broadcast to all of the neurons in the next layer. So, to use the network to solve a problem, we apply the input values to the inputs of the first layer, allow the signals to propagate through the network, and read the output values.

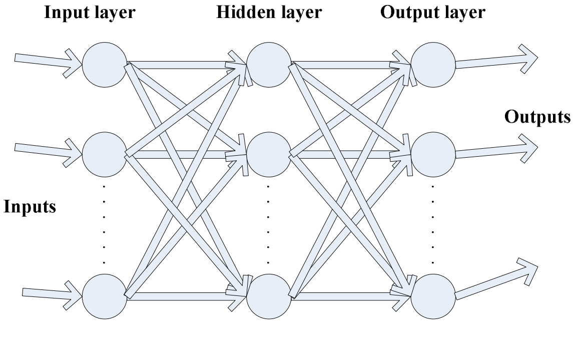


Figure 3 A Feed-Forward Nerual Network Model

Since the real uniqueness or 'intelligence' of the network exists in the values of the weights between neurons, we need a method for adjusting the weights to solve a particular problem. For this type of network, the most common learning algorithm is called Back Propagation (BP). A BP network learns by example, that is, we must provide a learning set that consists of some input examples and the known-correct output for each case. So, we use these input-output examples to show the network what type of behavior is expected, and the BP algorithm allows the network to adapt.

2) BP algorithm

The BP learning process works in small iterative steps: one of the example cases is applied to the network, and the network produces some output based on the current state of its synaptic weights (initially, the output will be random). This output is compared to the known-good output, and a mean-squared error signal is calculated. The error value is then propagated backwards through the network, and small changes are made to the weights in each layer. The weight changes are calculated to reduce the error signal for the case in question. The whole process is repeated for each of the example cases, then back to the first case again, and so on. The cycle is repeated until the overall error value drops below some pre-determined threshold.

3) Using Error Back Propagation (BP) algorithm to predict pixels

According to BP algorithm, we provide the value of the pixels which belong to the front fragment A_i as a learning set, and then use them to predict the next fragment's pixels value (Fig.4). We design some parameters as follows.

a) Nodes

The nature of the actual problem decides the network's input and output nodes which has nothing to do with the network performance. When the distance between pixels more than 5, the relevance of the pixels will be little, and in one of image regional area, color information wills not mutation. So we use *n* adjacent pixels to predict the current pixel. The choice of neighborhood pixels shows as Fig.5.

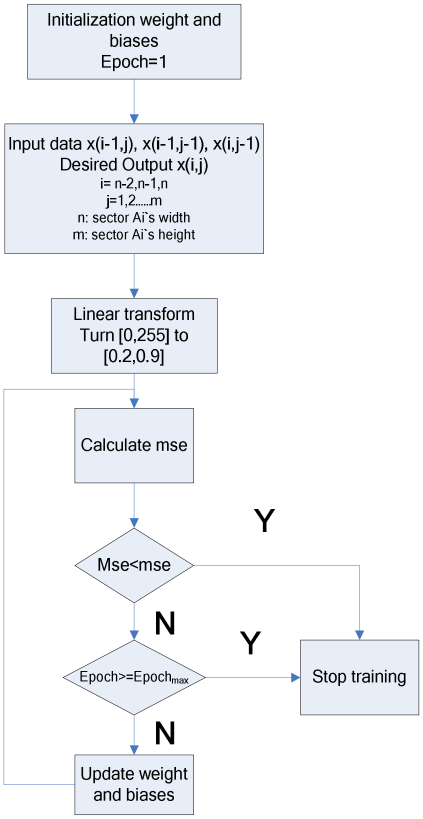


Figure 4 A training process flowchart

The two equation shows in the following how to predict the current pixel.

$$\hat{x}(i,j) = f(x(i+k, j+l)).(5)$$

$$e(i,j) = x(i,j) - \hat{x}(i,j).(6)$$

$$(k,l) \in \begin{cases} (-1,0),(-1,-1),(0,-1) & n=3 \\ (-2,0),(-1,0),(-1,-1),(0,-1),(0,-2) & n=5 \\ (-3,0),(-2,0),(-2,-1),(-1,0),(-1,-1),(-1,-2)(0,-1),(0,-2),(0,-3) & n=9 \end{cases} .(7)$$

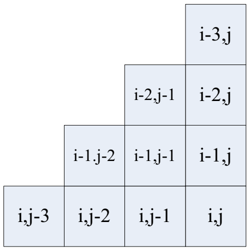


Figure 5 Neighborhood pixels used to predict the current pixel x(i,j)

Where $\hat{x}(i,j)$ is the predict value of the current pixel, $x(i,j)$ is the current pixel actual value, and $e(i,j)$ is the prediction error. So we have *n* neurons in the input layer and 1 neuron in the output layer.

b) Transfer Function

The classical BP algorithm uses sigmoid function as transfer function, and the output of the sigmoid function is in the dynamic range [0, 1]. The sigmoid function and its derivative are as follows.

$$f(x) = \frac{1}{1+e^{-x}}.(8)$$

$$f_x' = f(x)(1 - f(x)).(9)$$

Initialize the weighs and biases with random values, that makes the activities region of neurons in each layer can be broadly flat in the input space.

c) Input Sample

We use pixels in the last three rows of the front fragment A_i as training sample. In the test sample, we use pixels which belong to the fragment A_i and are adjacent to the first row of the back fragment A_j as the input of the network, and the output of the network is the predict pixels value of the first row of the back fragment A_j $\hat{x}(i,j)$.

d) linear transform

Through a simple linear transformation, we map the network input and output data [0,255] into the range [0, 1].

Experimentally, we found map the input and output data into the range [0.2, 0.9], the result of the prediction would be better.

$$x' = \frac{x}{255} \times (0.9 - 0.2) + 0.2 \text{ .(10)}$$

Where x is the original network input, and x' is the input which after the linear change.

When the network gets output, we should use "11" to map the result into the range [0,255].

$$y = \frac{(net - 0.2)}{0.7} \times 255 \qquad (11)$$

Where net is the output of the network, and y is the predictive value of pixels $\hat{x}(i, j)$.

4) Carving Algorithm

(1) The image is preprocessed to locate JPEG headers.

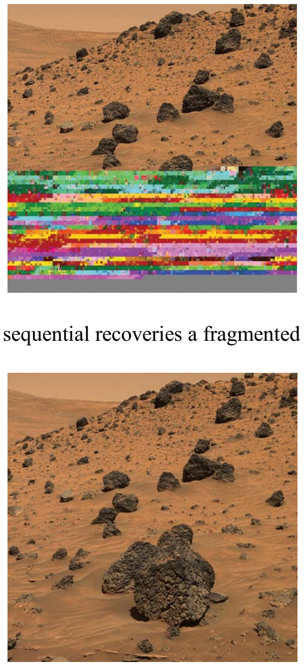
(2) A validator/decoder sequentially decodes a sector after the first sector at a time to see if the data of this sector belongs to JPEG. When errors occur, the validator/decoder discards this sector and begins to decode the next sector.

(3) The BP algorithm is used to determine that whether or not the given sector A_j follows A_i. If A_j not follows A_i, the discontinuity is detected, and then continues with step 2, decode the next sector A_{j+1}.

(4) If A_j follows A_i, merge sector A_j to data blocks set $\{A_0, A_1..., A_i\}$as A_{i+1}. If the file is not end, continues with step 2 to find whether the next sector follows the last block of the data set or not.

V. EXPERIMENTAL AND RESULTS

This section demonstrates the use of the proposed using BP algorithm to predict pixels by focusing on the JPEG file recovery, and then presents the experimental results for DFRWS 2006 test-sets. The DFRWS 2007 challenge test-sets contains JPEGs which have fragments in non-sequential order. Our method does not work when the fragments in non-sequential order.



(a) Incorrect sequential recoveries a fragmented image(MARS)

(b) The result of using our method to carve the same image

Figure 6 Example of reassembly of the Mars image which is collected from DFRWS 2006.

In our experiments we assumed that no fragments are missing and fragments are not randomly scattered but have multiple blocks sequentially stored. The set of JPEG file used for the experiments are collected from DFRWS 2006 test-sets. The DFRWS challenge states assume that all files begin on sector boundaries, fragmentation can only occur on sector boundaries and sectors are 512 bytes in size. Fig.6 provides an example of a JPEG file from the DFRWS 2006 test-set that was decoded incorrectly when doing sequential decoding.

In our experiments we attempt to carve the JPEG file based on this own internal contents. An example of comparing the value between the pixels which are calculated by BP algorithm using the pixels from the last sector as input data and the pixels which belong to the current given sector is shown in Fig.7.

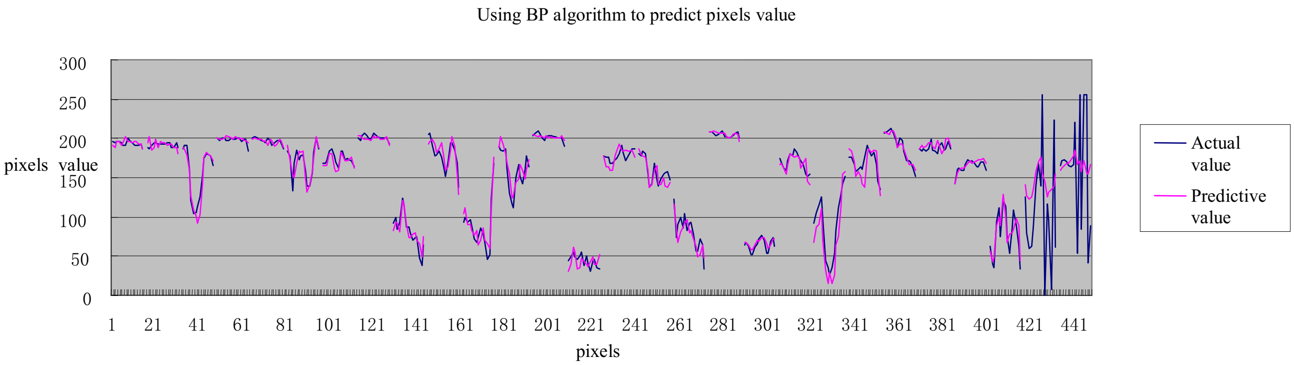


Figure 7 A graph of the value of the pixels result form our method to carve the image partly shown in Figure 6. As can be seen at value 418 the difference between two series gets larger from then on.

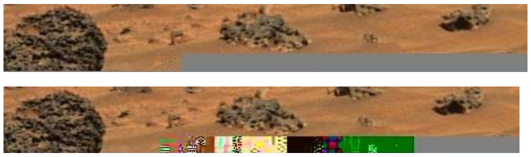


Figure 8 An example of an image corruption which can be revealed by decoding sectors one at the time.

Fig.8 shows this process. The image (of the Mars) is taken from the 2006 DFRWS at header offset 31533 sectors, the top part is shown after decoding 220 sectors while the bottom part was after decoding 221 sectors. The BP algorithm is then applied to the difference between the prediction pixels and the pixels have been decoded, then indicates an error. We can see before the 220th sector, the fragments of the image are in correct order, and the values of two series are almost the same before 418 (shown in Figure 7). After that, the difference between two series gets larger. It means that the 221th sector is not the one which adjacent with the 200th sector in the original but a discontinuity of the file.

TABLE I. BEST RECONSTRUCTION ALGORITHM FOR DATASET

Algorithm	*Total fragments recovered*	*False judge as a discontinuity*	*False judge as a adjacent fragment*	*Percentage recovered*
PM	5027	2302	67	67.503%
SOD	6062	1312	21	81.402%
MED	6509	916	6	87.404%
BP	6874	532	2	92.306%

The results of recover fragments using different algorithm are given in Table 1.We have 7447 fragments of a total 7586 fragments need to be computed. These fragments are from 9 JPEG images of which 7 are fragments, and these JPEG images are collected from the DFRWS 2006 challenge test-sets.

Taking one with another, we think that using BP algorithm to carve JPGE file is more useful than other three methods which are mentioned by Memon[8]. It can detect discontinuity more effective with lower "false positives". The false judgments are manually checked to find out the reason why BP algorithm can not properly categories them. When the value of the adjacent pixels changes more than 40, a false judgment may occur.

It should be noted that in order to improve the result of the experiment, we can use more pixels from last sector as sample data to be trained in BP algorithm, but the more data you train the more time it takes. It is also known that when the distance between pixels more than 5, the relevance of the pixels will be little. Experimentally we find using n (n=3) adjacent pixels to predict the current pixel and to indicates an error, the result and speed of the experiment are satisfactory.

VI. CONCLUSION AND FUTURE WORK

In this paper, we introduce the information about JPEG file carving and present a new method to carve JPEG file base on the JPEG file internal contents. After formulating the problem of reconstructing the fragmented JPEG images as a pixel prediction problem, we describes the technique that using BP algorithm to carve fragmented JPEG file. Finally, we compare different techniques that using in JPEG file carving method, and the result shows that our technique works better than others.

Currently, our carving method works well when a large number of fragments in sequential order. In order to support the situation that the fragments could be stored anywhere or in non-sequential order on the disk, we will develop our JPEG file carving method to be more useful and automatic in future work. It also should be more tested for improving its accuracy.

REFERENCES

[1] Pal and N. Memon, "The evolution of file carving," Volume 26, Issue 2, Digital Object Identifier 10.1109/MSP.2008.931081, Page(s):59 – 71, March 2009.

[2] B. Carrier, E. Casey, and W. Venema. DFRWS 2006 forensics challenge,URL http://dfrws.org/2006/challenge/ 2006.

[3] Golden G. Richard Ⅲ , Vassil Roussev, "Scalpel: A frugal, high performance file carver," Proceedings of the 2005 Digital Forensic Research Workshop. New Orleans, LA, 2005.

[4] S. Garfinkel. "Carving contiguous and fragmented files with fast object validation," Digital Investigation, 4(Supplement 1):2–12, Sept. 2007. DOI:http://dx.doi.org/10.1016/j.diin.2007.06.017.

[5] Michael Cohen, "Advanced carving techniques," Digital Investigation. Vol.4, Issues 3-4, pp.119-12, 2007.

[6] Michael Cohen, "Advanced JPEG carving," ICST, Brussels, Belgium, 2008

[7] N. Memon and A. Pal. "Automated reassembly of file fragmented images using greedy algorithms," *IEEE Transactions on Image Processing*, 15(2):385–393, Feb. 2006.

[8] A. Pal, K. Shanmugasundaram, and N. Memon. "Automated reassembly of fragmented images," In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo*, IEEE Computer Society, pages 625–628, Washington, DC, USA, 2003.

[9] K. Shanmugasundaram and N. Memon. "Automatic reassembly of document fragments via data compression," In *Digital Forensic Research Workshop*, 2002. http://isis.poly.edu/memon/publications/pdf/2002_Automatic_Reassembly_of_Document_Fragments_via_Data_Compression.pdf, 2008.01.14.

[10] K. Shanmugasundaram and N. Memon. "Automatic reassembly of document fragments via context based statistical models," In *Proceedings of the 19th Annual Computer Security Applications Conference*, 2003. http://www.acsac.org/2003/papers/97.pdf, 2008.09.12.

[11] A . Pal , T. Sencar , and N . Memon, "Detecting file fragmentation point using sequential hypothesis testing," Digit. Investig, 2008.05.15

[12] M. Karresand , N. Shahmehri, "Reassembly of Fragmented JPEG Images Containing Restart Markers," IEEE Computer Society Washington, DC, USA, Page(s): 25-32 , 2008