Width Extraction of JPEG Fragment via Frequency Coefficients Scale Similarity Measuring

Ye. Xu
Computer Science Department of
Hangzhou Dianzi University
HangZhou, China, 310085
xysmiracle@hotmail.com

Ming Xu
Computer Science Department of
Hangzhou Dianzi University
HangZhou, China, 310085
mxu@hdu.edu.cn

***Abstract*—the size field in encoded JPEG file header is very important to the image displaying, so it should be designated to avoid the problem of confusedly displaying. The current measure of constructing of a pseudo header does not deal with this issue very well. To address this problem, the paper presents an algorithm which uses luminance-DC-frequency-coefficients-scale differential algorithm to obtain the width of a header-missing JPEG fragment without using any original meta-data. We use both the algorithm and the construction of decipher prerequisites to get the value. The results of experiments have shown that they can obtain the width information and the location of 8*8 blocks sequence of header-missing JPEG fragment with resilience and accuracy.**

Keywords- frequency coefficients scale, size, data fragment, JPEG

I. INTRODUCTION

With the ever increasing exploitation of digital image for legitimate using, the need for more resilience and sophisticated JPEG file recovery products is also increasing. However, the ability to recover the broken JPEG fragmented data extracted from a digital storage medium without any file system information involves several algorithmic challenges. These challenges have two primary factors. Firstly, varying and complex situations may result in such a fact that JPEG fragmented data cannot be linked to an intact header. JPEG files are stored in encoded form to make sure that the compression and correction. This demands sophisticates decoders to decipher the data. With the lack of original information from file header, the process of deciphering will fail. Although [4] authors addressed the method of construction of a pseudo header to decipher the only head-missing image files, the problem that image display confusedly was not resolved. Secondly, in order to ensure the confidentiality of certain images, the size field of the header of an image has been tampered deliberately. So even if the data part of a JPEG image file is extracted and decoded correctly, the true contents of the image cannot be identified.

Recently, few researchers are concentrating on these facts of challenges. There are some situations such as the data part of the JPEG is fragmented or size field of the JPEG header is missing, overwrote partially or tampered deliberately may lead to the failure of image displaying. The information of image's size in header is a key factor for displaying. The absence of this information will lead to the problem of image 8*8 blocks' locations shifting, see Fig.2. (b). In this paper, we attempt to address one fundamental question, just give the data fragment of a JPEG file, and how we could get the correct location of 8*8 blocks sequence of a header-missing JPEG fragment (the size of data fragment of a JPEG) for displaying correctively and recover the data fragment of a JPEG effectively. In order to obtain frequency-based coefficients, our approach constructs some necessary prerequisites for data decoding first, then we utilize the luminance DC frequency coefficients scale differential algorithm to measure the width and height of data fragment of a JPEG file.

We provide an overview of our approach: section 2 provides a brief description of previous work on file carving and fragmented JPEG file recovery. The structural properties of JPEG file and our methodology are discussed in section 3. We present the results of experiments in section 4. In section 5, conclusion and future work are outlined.

II. RELATED WORK

Some technologies of fragmented JPEG files recovery have been proposed.

A novel method was presented by [1], which based on the information extracted from the outlines and the color contents of the images, without any knowledge of the final image.

In [2], authors defined the accurate carving as a multi-tier decision problem which seeks to validate or discard candidate byte string --"object"-- from the media to be carved quickly. Then validators for the JPEG format were discussed.

A sequential hypothesis testing---based recovery method of recovery of JPEG files is introduced in [3] , it identified the fragmented point of a JPEG file by sequentially comparing adjacent pairs of blocks from the start block to the fragmentation point .

In [4], authors addressed two problems related to JPEG file recovery. The first one tries to find a more efficient method to identify the next fragmented point of a JPEG file. And the second one focus on the recovery of file fragments which cannot be linked to the existing image headers or for which there are no available image headers. Authors addressed the construction of a pseudo header needed for recovery of stand-alone file fragments. Although many

978-1-4244-5824-0/$26.00 ©2010 IEEE

authors addressed it's important to build pseudo headers for the files, they did not pay much attention to the size field of image. The size information of an image in header is a key factor for image displaying. The absence of this information will lead to the problem of the locations shifting of the image 8*8 blocks. It caused the problem of displaying and identifying of the image.

[5] presented a method to reassemble fragmented JPEG images containing Restart markers. It considers the situation that in the case of a missing original file header; the authors come up with a method that creation of an artificial JPEG image header. Although authors addressed method for calculating width of the image, there were many limitations. Authors assume there is at least one vertically oriented line in the image. The method will fail when the vertical lines do not exist in the image or when there are only some horizontal oriented lines exist in the image.

III. METHODOLOGY

A. *JPEG file format*

1) Basic storage structure

Each JPEG file contains a header section and a data section, which is called scan. The header section consists of some markers which contain tables and other information needed to decode the image data in the scan. The majority of the markers in the header are very important to decoding of a stand-alone image data. Considering only baseline JPEG/JFIF images, the most common JPEG encoding method used in most cameras, the most important markers for decoder/encoder are listed below:

- APP0 marker: it is made up of the identification, resolution and thumbnail information.
- DQT maker: it specifies one or more quantization tables.
- SOFn maker: indicates that this is a baseline DCT-based JPEG, and specifies the width, height, number of components and component sub-sampling.
- SOS maker: this marker specifies which slice of data it will contain, and is immediately followed by entropy-coded data.
- DHT maker: the segments which begin with a DHT marker specify one or more Huffman tables.
- Restart (RST) markers: they are used the scan is interrupted at regular intervals^2 by a marker 0xFFDx, where xi+1 = (xi + 1) mod8; x0 = 0.

2) JPEG decoding

Most image software use the simpler JFIF format when creating a JPEG file. Typical JPEG decoders have four main steps. Firstly, quantized transform are right ordered via a lossless entropy coder's decoding and went through re-zigzag scanning within the transformed block. Then the ordered coefficients are re-quantized. Thirdly, Two-dimensional DCT is applied to each of the blocks to obtain coefficients in standard color space which transform from coefficients in frequency domain. Finally, the non-overlapping blocks of size 8*8 pixels are laid out at their proper locations on the image for display correctly.

The needed information to decode an image from header can be divided into five parts. They are:

- The width and height of the image specified in number of pixels.
- The 8*8 block quantization tables used during decompression.
- The type of chroma sub-sampling in composition.
- The order of color components in a MCU.
- A set of valid Huffman tables.

B. *Recover algorithm*

1) Prerequisites

The fragmented data of a JPEG file presented in this paper is incomplete; it will cause serious error damage due to the error propagation. Using of restart markers (RST marker) in the JPEG standard provides the resynchronization function for error handling. All erroneous restart markers are corrected and rearranged in the correct order. We therefore assume that relevant data fragment have already contained the RST markers.

We utilize four in five parts mentioned in 3-A-2 as the requisites of a pseudo header.

Firstly, the starting block location is identified by using the sequence of RST markers. Secondly, because of different quantization tables only affect the visual quality; we can use standard JPEG quantization tables [6] for the data fragment of JPEG file. Thirdly, we can only utilize the Huffman tables from the JPEG standard or the adjacent recovered images, if the data fragment of JPEG file use non-standard / non-adjacent tables, a brute force model is feasible. Fourthly, because there are only 12 different possible combinations of sampling factor settings, it is can be easily determined by trying all settings. Finally, the arrange order of color components can be determined by simple appropriate Huffman table look-up operation [4].

We construct a pseudo header through the above information to decode the data fragment correctly.

2) Luminance DC frequency coefficients scale-based differential algorithm

The differential algorithm is usually used to process images, which are based on gray scale. It is a very effective focusing evaluation function in the field of digital image processing, although it has a very simple form. It utilizes the sum of absolute difference values between adjacent Pixels as the evaluation of focus effects. However, since the changes between image pixels are so slight, thus the difference between values which calculated by the gray-scale differential algorithm is too small to meet the requirements point of monotony; its results are not satisfied. So luminance DC frequency coefficients scale-based differential algorithm is proposed.

The formula (1), where f(x, y) is the luminance DC frequency coefficients function in the two-dimensional coordinate system. And f(x, y-1), f(x-1, y) are the adjacent luminance DC frequency coefficients value in adjacent direction.

$$F(x,y)=\sum_{x,y}\{|f(x,y)-f(x,y-1)|+|f(x,y)-f(x-1,y)|\} \quad (1)$$

3) Calculate the size of fragmented image

We use luminance DC frequency coefficients scale differential algorithm to measure the width and height of a fragmented data of JPEG file. The DC luminance value is the most important factor to the JPEG image. It contains the profiles of the objects in the image, so it can be used to reveal a frequency-coefficients-scale vision of the image.

The fact of this method is that the scan of a JPEG image is started at the top-left corner and precedes one horizontal line at a time downwards. Therefore the adjacent two lines will be reflected as a high similarity of the DC values. The rate of change of the similarity values calculated when the adjacent two rows move in parallel indicates whether the width value is correctly got or not.

Before describing the algorithm, we'll give a series of definitions.

- Definition 1 (two-dimensional coordinate (x, y)): we suppose (x, y) are the two-dimensional coordinate which represent the location of a luminance block. A luminance block is a minimum unit in our algorithm, and it consists of 8*8 frequency coefficients. For example: let (4, 5) denotes the coefficient of the 4-th row and 5-th column of data matrix.
- Definition 2 (Similarity (W)): in order to derive a differential algorithm-based similarity robustly from rows of coefficients, we propose the function for calculating the similarity value in specified W. For example: Similarity (60) returns the similarity value when the width is 60 blocks.

$$Similarity(W)=\frac{\sum_{y=1}^{R}Row\_Diff(y)}{R} \quad (2)$$
$$(0\leq W\leq Total\_Num)$$

- Definition 3 (Row_Diff (y)): a function Row_Diff(y) indicates the average of differential values of all blocks in the y-th row. The variable W is our assumption for the width of image. For example: Row_Diff(9) returns the average of differential values of all blocks in the ninth row.

$$Row\_Diff(y)=\frac{\sum_{x=1}^{W}Each\_Diff_{y}(x)}{W} \quad (3)$$

- Definition 4 ($Each\_Diff_{y}(x)$): we define Each_Diffy(x) function as differential of each block in specified coordinate. By the formula (4), where f(x, y) is the DC luminance function in the two-dimensional coordinate system. And f(x, y-1), f(x-1, y) ... are the adjacent DC luminance values in up, down, left, and right direction.

$$Each\_Diff_{y}(x)=\sum_{x,y}\{|f(x,y)-f(x,y-1)|+$$
$$|f(x,y)-f(x-1,y)|+|f(x,y)-f(x+1,y)|+$$
$$|f(x,y)-f(x,y+1)|\} \quad (4)$$

- Definition 5 (R value): if blocks of data are decoded correctly, we can get the value of R (5) by dividing the total number of decoded blocks (Total_Num) by width(W).
  For example: R = 10/5 indicates that R value is obtained from divide 10 by 5.

$$R=\frac{Total\_Num}{W} \quad (5)$$

- Definition 6 (Min_Avg (sim [])): the function Min_Avg (sim []) is used to calculate the average interval value between these array indexes which were chosen by the array sim[] to get the minimum values in different local area. The sim[] array contains Similarity(W) values in specified width. For example: if Similarity (x1), Similarity (x2), Similarity (x3), Similarity (x4), Similarity (x5) are 5 minimum values in different local area, the function will return the average interval value avg = ((x2-x1) + (x3-x2) + (x4-x3) + (x5-x4))/4.
- Definition 7 (Proper_Height (W_{proper})): the function Proper_Height (W_{proper}) returns the height of the image by passing into a W_{proper} value. If the W_{proper} satisfies equation (6), the height value returns by the equation (7). Else we can get the height value by rounding up the value calculated by equation (8).

$$Total\_Num\%W_{proper}=0 \quad (6)$$

$$H_{proper}=\frac{Total\_Num}{W_{proper}} \quad (7)$$

$$H_{proper}=\left\lceil\frac{Total\_Num}{W_{proper}}\right\rceil \quad (8)$$

Algorithm 1: size extracting algorithm

```
Input: S: sequence of DC luminance values
       Total_Num: the length of the sequence
Output: L: Size of the fragmented image.
    Step 1: Initialize Sim = $\varnothing$ as the arrays which store the values of Similarity (W) in the specified range of W;
    Step 2: For W = 1 $\rightarrow$ Total_Num: Sim [W] = Similarity (W);
    Step 3: $W_{proper}$ = Min_Avg (Sim []);
    Step 4: $H_{proper}$ = Proper_Height ( $W_{proper}$ );
    Step 5: return (Size of the image fragment);
```

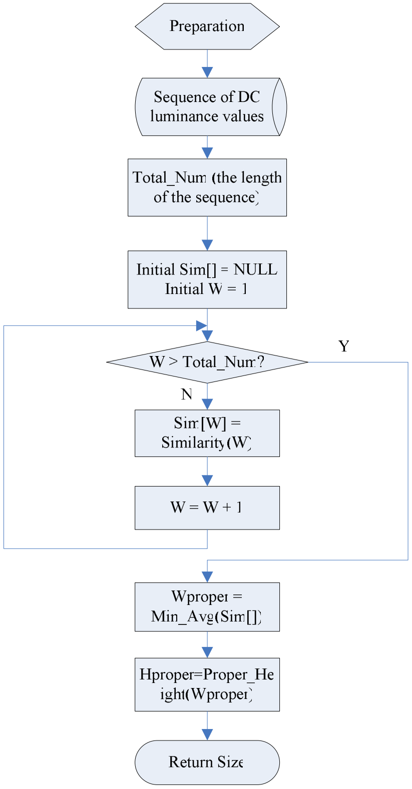


Figure 1. Size extracting algorithm

Firstly, we'll do some initialization work in step 1. And then, we calculate Sim[W] based on specified W value recursively. This is show in step 2. Thirdly, the proper W value is calculated by Min_Avg (Sim []) function and the proper H value is obtained from Proper_Height() function. This is shown in step 3 and step 4. Finally, we can get the size value as shown in step 5.

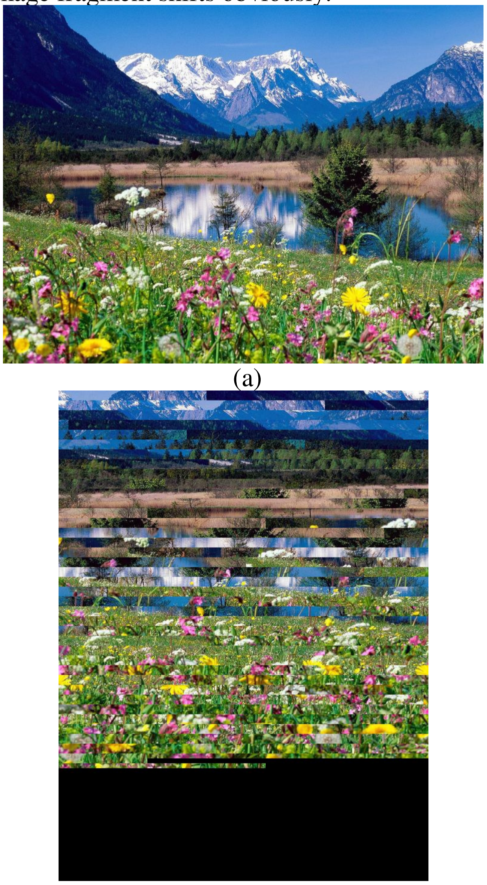
IV. EVALUATION AND EXPERIMENTS

In our experiments, the experimental material consists of representative images download from the internet randomly, with the size of 800*600 each. We take a landscape picture for example here. It is captured by the digital camera. And the various colored objects are mixed together, showing a gradual change in color with a rich and harmonious tone. After recording the original size of image, we erased the header and some horizontal lines' blocks of the JPEG image, only remained incomplete scan part (data part) of it.

We must make sure all of the prerequisites (mentioned in 3.B.1) were prepared, before we got the sequences of DC luminance values from decoded blocks of data. These prerequisites consisted of a pseudo header for decoding the blocks of data correctly.

The proposed techniques and their implementation target the machines running under the Windows XP (SP1, SP2) operating system.

In Fig.2 (a) is one original JPEG image and (b) is the data fragment after (a)'s header and blocks were erased. It is a missing blocks image, in which 870 blocks are missed from the beginning of the first row. Because the size filed of pseudo header is filled with a random value, it is clear in (b) that the image fragment shifts obviously.

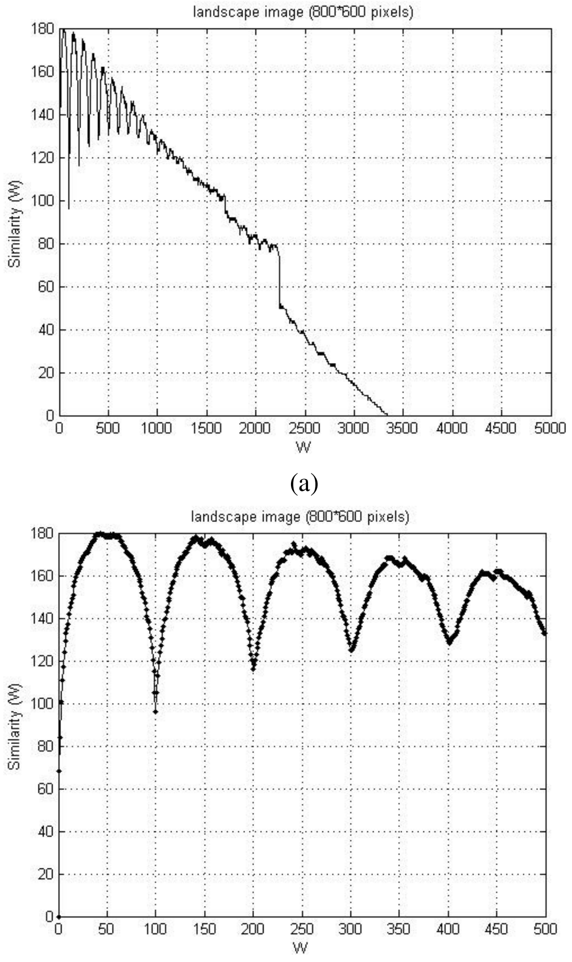


(a)

(b)

Figure 2. (a) original image, (b) display of blocks missing image with wrong width.

Fig.3 gives the results of our size extracting algorithm. (a) gives all values of the data fragment. The reason why Similarity (W) value trend towards 0 is that the bigger W is, the smaller R is. During to the limited space here, we show the results of the values in the range from 1 to 500 too, see (b). The X-axis denotes different W values, and Y-axis denotes the different equation 1 values while W is assigned specific value. We can observe that the value of interval of troughs is fixed. So the interval value is obtained as the candidate of right width of image. The reason is that the closer to the correct width, the smaller the Similarity (W) values will be. Taking (b) for example, the interval value we get is very close to the original width of image (100≈800/8=100).



(a)

(b)

Figure 3. (b) Similarity(0,W) VS. W (W ∈ [1,500]). Value of interval of troughs ≈ 100.

After W and H values were got, we arranged the whole blocks with the rule that W blocks in one row. As the results output demonstrated in Fig.4, we can obtain the location of 8*8 blocks sequence of a header-missing JPEG fragment without using any original meta-data.

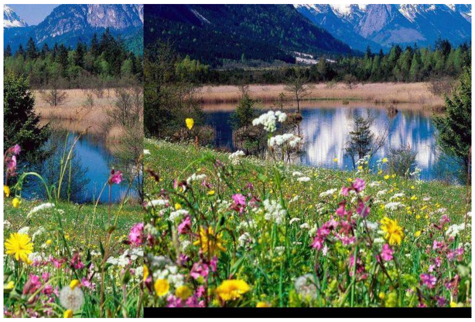


Figure 4. the outcome of our experiment

V. CONCLUSION AND FUTURE WORK

In this paper, we used differential algorithm to measure the similarities between rows of DC luminance values in specified width (W) and took the interval of min_similarity values as the candidate of width of the fragment data of image. An algorithm has been proposed to calculate the width of a header-missing JPEG fragment without using any original meta-data. We used representative fragment data of JPEG images to test the resilience of our method. Our experiment has shown that our method has a certain degree of accuracy.

In the terms of the future work, we will do more experiments to verify the reliability of our method, and shift our attention to the situations such as the JPEG files that without containing RST marker are fragmented, broken, and partially modified. Our algorithm should be improved to adapt to this situation.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of Zhejiang Province of China under Grant No. Y1090114. Thanks very much.

REFERENCES

[1] Francesco Amigoni, Stefano Gazzani, Simone Podico, “A method for reassembling fragments in image reconstruction”, 2003 International Conference on Image Processing, Institute of Electrical and Electronics Engineers Computer Society, 2003, p 581-584.

[2] Simson L.Garfinkel, “Carving contiguous and fragmented files with fast object validation”, Digital Investigation, Elsevier Ltd, 2007, p 2-12.

[3] Anandabrata Pal, Husrev T.Sencar, Nasir Memon “Detecting file fragmentation point using sequential hypothesis testing”, Digital Investtigation, Elsevier Ltd, 2008, p S2-S13.

[4] Husrev T.Sencar, Nasir Memon, “Identification and recovery of JPEG files with missing fragments”, Digital Investigation, Elsevier Ltd, 2009, p S88-S98.

[5] Martin Karresand, Nahid Shahmehri, “Reassembly of Fragmented JPEG Image Containing Restart Markers”, Proceedings-4th Annual European Conference on Computer Network Defense, Inst. Of Elec. And Elec. Eng. Computer Society, 2008, p 25-32.

[6] Jesse D. Kornblum, “Using JPEG quantization tables to identify imagery processed by software”, Digital Investigation, Elsevier Ltd, 2008,pS21-S25.