FRAGMENTED JPEG FILE RECOVERY USING PSEUDO HEADERS

Yanbin Tang, Zheng Tan, Kam-Pui Chow, Siu-Ming Yiu, Junbin Fang, Xiamu Niu, Qi Han, Xianyan Wu

Chapter 13

FRAGMENTED JPEG FILE RECOVERY USING PSEUDO HEADERS

Yanbin Tang, Zheng Tan, Kam-Pui Chow, Siu-Ming Yiu, Junbin Fang, Xiamu Niu, Qi Han and Xianyan Wu

Abstract Many techniques have been proposed for file recovery, but recovering fragmented files is still a challenge in digital forensics, especially when the files are damaged. This chapter focuses on JPEG files, one of the most popular photograph formats, and proposes techniques for recovering partially-damaged standalone JPEG fragments by reconstructing pseudo headers. The techniques deal with missing Huffman tables and sub-sampling factors, estimate the resolution of fragments, assess the image quality of JPEG files with incorrect quantization tables, and create quantization tables that are very close to the correct quantization tables in a reasonable amount of time. Experiments with real camera pictures demonstrate that the techniques can recover standalone fragments accurately and efficiently.

Keywords: JPEG fragments, file recovery, pseudo headers

1. Introduction

Operations such as creating, deleting, editing and appending files cause the fragmentation of user and system files, especially on hard drives that are filled to capacity and flash storage devices that employ wear-leveling mechanisms. Depending on the new file allocation rules of a filesystem [1], deleted files are overwritten by newly-created or edited files. JPEG is one of the most popular image file formats used in computers, digital cameras and smartphones and on the Internet. JPEG image files often constitute crucial evidence in criminal investigations. However, if the underlying filesystem is corrupted, it is difficult to recover files, especially when the files are fragmented, destroyed or incomplete.

This chapter proposes techniques for recovering partially-damaged standalone JPEG fragments by reconstructing pseudo headers. The techniques deal with missing Huffman tables and sub-sampling factors, estimate the resolution of fragments, assess the image quality of JPEG files with incorrect quantization tables, and create quantization tables that are very close to the correct quantization tables in a reasonable amount of time. Experiments with real camera pictures demonstrate that the techniques can recover standalone fragments accurately and efficiently.

2. Related Work

Many researchers have focused on file recovery and/or file carving. However, most of the techniques implemented in digital forensic tools (e.g., Encase, FTK and The Sleuth Kit) recover complete files by interpreting the filesystems in storage media. The vast majority of files are organized using specific structures or are encoded by specific algorithms. If a deleted file has been overwritten partially, it is difficult to recover the file fragments by exclusively analyzing filesystem data. The recovery of damaged encoded files and compressed files is also a major challenge. In such instances, it is necessary to analyze the file structures, obtain the necessary information and apply specific algorithms to read and display the damaged file content.

In the context of JPEG files, researchers have formulated the problem of reassembling JPEG fragments as a Hamiltonian path problem and have developed several greedy heuristic algorithms to identify the optimal path [8, 11, 13]. Garfinkel [3] has proposed a fast object validation technique to recover bi-fragmented files by exhaustively analyzing all possible combinations of data blocks between an identified file header and footer. Karresand and Shahmehri [6] have developed a method for reassembling a special type of JPEG file with “restart markers” that handles bit stream errors due to file corruption or unreliable transmission (however, not all JPEG files have such restart markers). Pal et al. [10] have proposed a sequential hypothesis testing algorithm that reassembles fragmented files by detecting JPEG file fragmentation points. However, if a file has been damaged or corrupted, the methods mentioned above reconstruct the file incorrectly or are simply unable to find the correct next fragmentation point.

Sencar and Memon [12] have proposed a technique for recovering JPEG files with missing fragments. The technique leverages the fact that image decoding information and content are stored separately in the JPEG header and the body. However, the technique is too general to construct pseudo headers. Inspired by Sencar and Memon's work, this research proposes an approach for constructing pseudo headers piece by piece.

Xu and Xu [18] have proposed an algorithm that estimates the width of JPEG pictures by measuring similarities between the differences of DC luminance coefficients in each row of an estimated picture width, which involves a brute force search from one up to the real width. When the estimated width is equal to the real width, the best (smoothest) features are obtained throughout the picture along with the closest similarity. However, the time consumption of this algorithm is as high as $(w \times h)^2$ where w is the width of the test image and h is the height.

Other researchers have attempted to predict the quantization tables of JPEG files [4, 17]. However, the technique only work for files with extended IJG standard quantization tables [5]. The technique estimates the similarity between coefficient differences by simulating the JPEG decoding and encoding processes with specific quality factors. However, the technique is very time consuming because it processes coefficients using discrete cosine transformation, quantization, dequantization and inverse discrete cosine transformation repeatedly for quality factors ranging from one to 100 to obtain the best result. The time requirements would be even more for test pictures with high image quality.

3. JPEG Background

A sound JPEG file has two parts, the header and the data (body). The header contains nearly all the configuration information used by the decoding process. The data part contains pure encoded binary data with an ending marker that indicates the end of the data section. The JPEG standard defines four encoding modes: lossless, sequential (or baseline), progressive and hierarchical. The sequential mode, which is the default and most commonly used JPEG mode, is the focus of this work.

The JPEG encoding algorithm transforms the color space (e.g., RGB) of an input image to the color components YCbCr. The three color components are divided into minimum coded units (MCUs), which are usually non-overlapping blocks of 8×8 pixels. The specific number of pixels per color component in a minimum coded unit is determined by the sub-sampling factor in the JPEG file header. The discrete cosine transform is used to transform the color pixels to frequency domain coefficients. The low-frequency coefficients are concentrated in the top-left portion of a minimum coded unit matrix while the high-frequency coefficients are in the bottom-right portion. Because humans are not sensitive to high-frequency information, after the quantization process, many of the high-frequency coefficients are rounded to zero or small numbers, thus achieving lossy compression.

The first element of a minimum coded unit is the direct component (DC) coefficient while the other elements are the alternating component (AC) coefficients. The DC coefficient is coded by differential pulse code modulation, which means that the differences between the DC coefficients of blocks are stored. AC coefficients are coded by run-length coding after zigzag scanning. The run-length coding can indicate the number of elements in a minimum coded unit matrix with an end of block (EOB) code. In the final step of the JPEG encoding process, all the coefficients are encoded by a variable bit length entropy coder that employs Huffman coding. The JPEG decoding process is performed by performing the steps in reverse order.

3.1 Essential Configurations in JPEG Headers

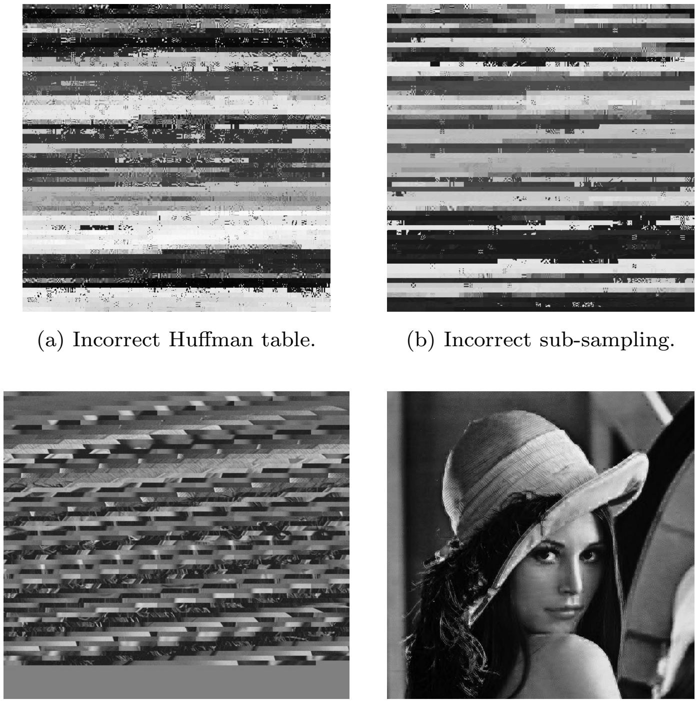
As described above, several processes and configurations are involved in JPEG encoding and decoding. To recover the content of a standalone JPEG fragment, it is necessary to construct a pseudo header to decode the fragment and then display it [12]. In the baseline JPEG encoding mode, four essential configurations are required to reconstruct a pseudo header: (i) Huffman code table; (ii) chrominance sub-sampling factor used in the minimum coded units; (iii) image resolution; and (iv) quantization table.

Figure 1 shows failed decodings of the `lena.jpg` file produced by incorrect components in the file header. Solutions for addressing the problems are discussed in Section 4.

3.2 Synchronization Point

The Huffman code as used in a JPEG file has the self synchronization property [7]. In other words, there is a synchronization point after which subsequent bits can be decoded correctly in any situation, even when starting from an unknown position.

In fact, the cost is only a few bits of error until the synchronization point is reached. An experiment was conducted with 700 pictures captured by several popular digital cameras and smartphones. The data part of each picture was truncated five times with a random size. Since DC coefficients are coded by differential pulse code modulation before the entropy encoding, there is only one DC coefficient in each color component of a minimum coded unit. Thus, a DC coefficient may occupy more minimum coded units before the synchronization point than an AC coefficient. The number of incorrect differential DC (dif_DC)



(a) Incorrect Huffman table.

(b) Incorrect sub-sampling.

(c) Incorrect resolution.

(d) Incorrect quantization table.

Figure 1. Versions of `lena.jpg` with incorrect components in the file header.

coefficients of luminance were counted to investigate the JPEG synchronization property. The results shown in Figure 2 demonstrate that more than 50% of the pictures reach the synchronization point after two minimum coded units and all the pictures become correct within 30 minimum coded units. Compared with the total number of minimum coded units in an image, the number of incorrect minimum coded units is negligible. Therefore, once the synchronization point is approached, all the entropy encoded data streams can be decoded successfully.

4. JPEG File Recovery Methodology

This section describes the proposed methodology for reconstructing pseudo headers. First, the impact of the four essential configurations (Huffman code table, chrominance sub-sampling factor, image resolution and quantization table) are discussed. Following this, techniques for predicting these configurations are presented.

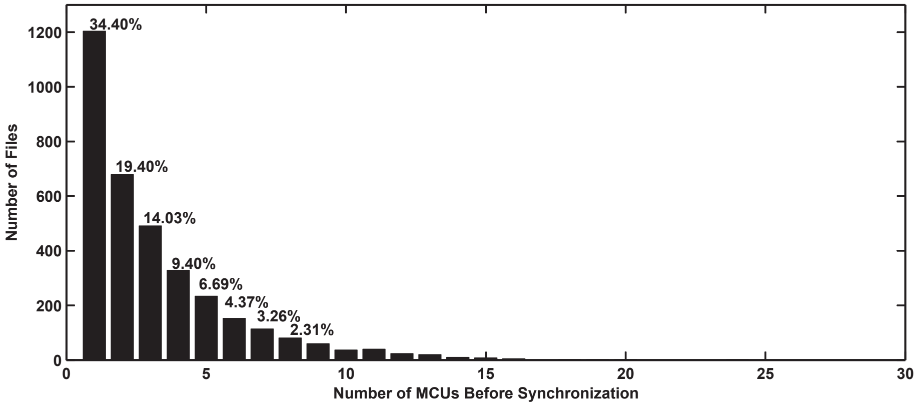


Figure 2. Number of incorrect MCUs before the synchronization point.

4.1 Huffman Table and Sub-Sampling Factor

In a baseline JPEG file, a mismatched Huffman table or chrominance sub-sampling factor may result in failed decodings as shown in Figures 1(a) and 1(b). Estimating a Huffman table from an encoded data stream is a major challenge. Sencar and Memon [12] have analyzed if Huffman tables from two fragments are identical considering a unique n-bit pattern. The probability of finding the bit pattern from an m-bit fragment is as low as $m/(2^n)$; when $n = 15$, the probability is around 3×10^{-05}. In order to reduce implementation costs, however, most digital camera manufacturers prefer to use the standard Huffman table [5], which was developed based on the average statistics of a large set of images. In fact, Karresand and Shahmehri [6] found that 69 of 76 popular digital cameras use the standard Huffman table. Consequently, the JPEG files considered in this work are assumed to use the standard Huffman table.

As specified in the JPEG standard, AC coefficients are encoded by run-length coding in interleaved order, which is defined by the sub-sampling factor. Also, an end of block (EOB) code exists at the end of each encoded minimum coded unit to identify the end of the current block. Thus, for each minimum coded unit, if a decoding is performed using an incorrect sub-sampling factor, the total size of the AC coefficients might go below or above the stipulated value (e.g., 63 for an 8×8 minimum coded unit block). Moreover, the number of sub-sampling factors used are very limited; the most popular settings in digital cameras are 4:1:1, 2:1:1 and 1:1:1. Therefore, by enumerating the limited sub-

Table 1. DC and dif_DC coefficients in MCU units from (0,0) to (7,5).

DC	**0**	**1**	**2**	**3**	**4**	**5**	**dif_DC**	**0**	**1**	**2**	**3**	**4**	**5**
0	46	46	46	46	47	47	0	46	0	0	0	1	0
1	61	60	59	59	59	60	1	1	-1	-1	0	0	1
2	60	59	60	61	**5**	-13	2	0	-1	1	1	**-56**	-18
3	59	59	59	61	**9**	-8	3	-1	0	0	2	**-52**	-17
4	59	59	59	61	**5**	-4	4	-1	0	0	2	**-56**	-9
5	59	59	59	61	**-3**	-7	5	23	0	0	2	**-64**	-4
6	59	59	59	55	-14	-5	6	79	0	0	-4	-69	9
7	59	59	53	-37	-9	-4	7	53	0	-6	-90	28	5

sampling factors and determining the total size of the run-length coded AC coefficients, it is possible to discover the correct configuration.

4.2 Image Resolution

If the width or height are mismatched, the decoded picture might appear to be filled with random data as shown in Figure 1(c). This chapter proposes a new technique to estimate the image resolution by analyzing the distances between similar dif_DC chains of luminance.

After the discrete cosine transformation, the first top-left DC coefficient holds the average intensity of the minimum coded unit block; it contains high energy because it is the most important coefficient. As specified in Property 1 below, real-world images have smooth transitions in object intensity and color; in other words, a specific profile or significant change always extends to adjacent areas. Thus, the DC coefficients can represent the luminance profile of a picture and the values are distributed sparsely to reflect the exact image pattern. Since the dif_DC coefficients are computed as the differences between the DC coefficients of adjacent encoding blocks from left to right, the dif_DC coefficients hold the change patterns in a compact form and concentrate on small values.

Property 1. Smoothness Phenomenon: *Generally, the luminance intensity and chrominance of a captured picture changes gradually over a localized area.*

Table 1 shows the DC and dif_DC coefficients in the minimum coded units (0,0) to (7,5) of Figure 3. The object intensity and color change significantly in the chimney region in Figure 3. Accordingly, the DC coefficient values drop significantly in column 4 from row 2 to 6 (shown in boldface). The dif_DC coefficients exactly represent this profile as shown in boldface on the right-hand side of Table 1. Comparing the DC

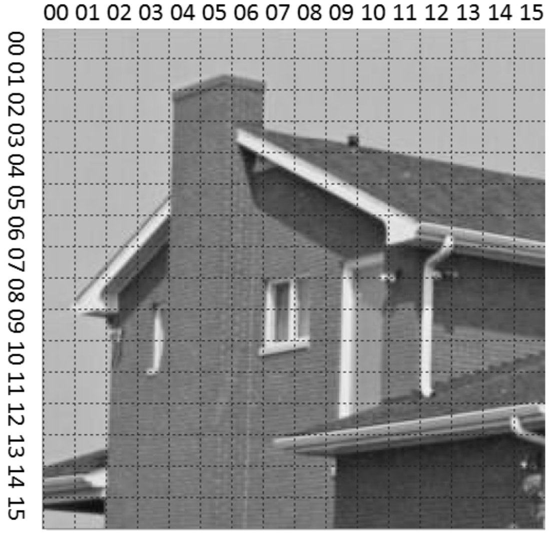


Figure 3. JPEG image of a house with the minimum coded units shown.

and dif_DC coefficients reveals that the dif_DC coefficients have smaller values for adjacent similar blocks.

The problem of estimating the image resolution can be transformed to finding the distance between two vertically adjacent minimum coded unit blocks. As demonstrated above, when there is a significant change, the pattern has a high probability of being extended to an adjacent area. As a result of improvements in image quality, the pattern would be covered by more than one minimum coded unit block in a high-resolution picture. In order to find the significant patterns of the dif_DC coefficients, a unique dif_DC chain is chosen as a sample chain. This sample chain begins with a high contrast dif_DC coefficient. Upon comparing nearby data areas, the most similar chains are found using cosine similarity:

$$cos(A,B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{1}$$

Accordingly, the distance between the most similar dif_DCs chains would be the width of the image.

In order to reduce noise and improve performance, m sample chains are chosen in each fragment and then the top k most similar chains are found. After evaluating the distances between the top $m \times k$ dif_DC chains, the distance value with highest frequency is chosen as the candidate width. Experiments revealed that the size l is less than 20, k

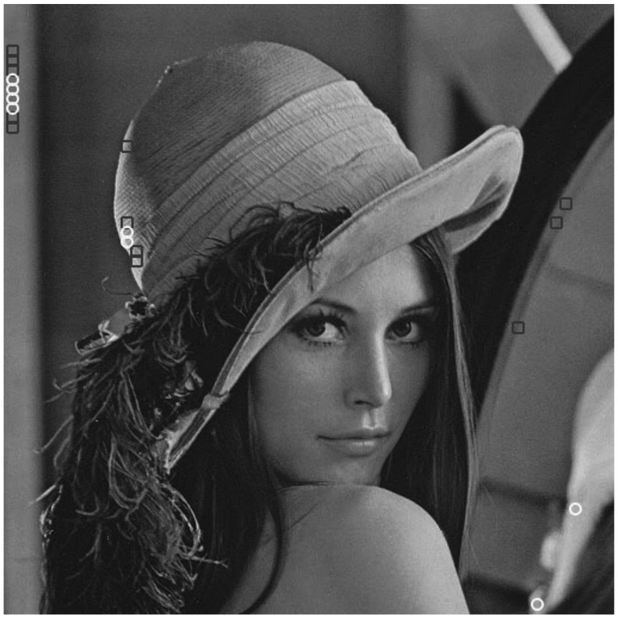


Figure 4. Image of `lena.jpg`.

is as small as five and m is smaller than ten, depending on the size of examined JPEG fragments.

In fact, human-vision-detectable vertical luminance changes are not necessarily required for fragments. As a result of improvements in image quality, even a small (significant) spot might contain several minimum coded unit blocks. In the case of the `lena.jpg` file with 512 × 512 resolution in Figure 4, the starting minimum coded units of the chosen sample dif_DC chains are indicated as circles and the evaluated similar chains as boxes. In this case, some of the sample chains also evaluated as the most similar chains. The distance values for points in the top-left portion and on the hat of `lena.jpg` have the highest frequencies. In other words, the highest frequency distance value is exactly equal to the width of the picture.

4.3 Quantization Table

This work focuses on original (unmodified) pictures. Unlike the situation involving image forgery detection in double quantized files[2], histogram patterns are inadequate for original photos. Also, when testing different quantization tables for files, distorted images do not have an impact on features such as Gaussian noise or Gaussian blurring. This section presents an image quality assessment (IQA) algorithm called Blocking_IQA for measuring the quality of JPEG files with incorrect quantization tables. Also, a quantization table prediction algorithm called QT_Blocking is described.

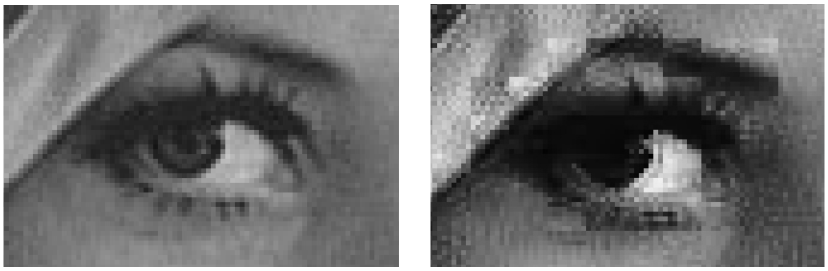
Image Quality Assessment (Blocking_IQA). Image quality assessment attempts to measure the quality of distorted pictures caused by restoration, transmission, compression and so on. Depending on the availability of a reference image, objective image quality assessment metrics can be classified into three types: (i) full reference (FR); (ii) no-reference (NR); and (iii) reduced-reference (RR). The full reference metrics, such as peak signal-to-noise ratio (PSNR) and SSIM [15], evaluate picture quality using the original picture as the reference. In contrast, no-reference metrics, such as jpeg_quality_score [16] and BRISQUE [9], predict the quality of a test picture only using information from the test picture. In fact, as elaborated by Wang et al. [16], it is still a major challenge to find a no-reference metric that can handle every distorted picture.

As defined by the JPEG standard decoding process, if a quantization table mismatch exists, color coefficients are increased or decreased after the dequantization process during decoding. Thus, the smooth features of a decoded picture with regard to luminance and color are seriously degraded, especially with blocking at the boundaries of JPEG minimum coded units as stated in Property 2:

Property 2. Blocking Phenomenon: *For a non-artifact picture, if a mismatch exists in the quantization table, there is visual degradation of luminance and chrominance and an impact on blocking at the boundaries of minimum coded units.*

Figure 5(a) shows a zoomed image of an eye in `lena.jpg` while Figure 5(b) shows a version of the same picture with an incorrect quantization table. Note that Figure 5(b) has darker luminance and worse chrominance than the original picture (Figure 5(a)). Most importantly, there is serious blocking around the eye, which is located at the boundaries of minimum coded units.

In order to analyze the image quality of pictures with incorrect quantization tables, the blocking property is evaluated by comparing the color differences between the internal and boundary blocks of minimum coded units. First, CIE76 (or Euclidean distance)[14] is used to measure the



(a) Original file.

(b) Incorrect quantization table.

Figure 5. Blocking phenomenon when zooming into an eye in `lena.jpg`.

color difference between two adjacent pixels:

$$CIE_i = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(x_i - x_{i+1})^2} \tag{2}$$

where $i \ mod \ 8 = \{6, 7\}$.

In the vertical direction, two CIE color differences are evaluated, one for the last two columns of each minimum coded unit and the other for two columns of two nearby minimum coded units. The blocking phenomenon ΔCIE is measured by comparing two adjacent CIEs as follows:

$$\Delta CIE = |CIE_i - CIE_{i+1}| \tag{3}$$

The estimated image quality score in the vertical direction $BlockingIQA_v$ is the average value of all the corresponding ΔCIE values:

$$BlockingIQA_v = \overline{\Delta CIE} \tag{4}$$

The image quality score in the horizontal direction $BlockingIQA_h$ is evaluated similarly. The final image quality score $BlockingIQA$ is the average of $BlockingIQA_v$ and $BlockingIQA_h$:

$$BlockingIQA = \frac{1}{2}(BlockingIQA_v + BlockingIQA_h) \tag{5}$$

A smaller $BlockingIQA$ score corresponds to better image quality and greater smoothness at the boundaries of decoded minimum coded units.

Quantization Table Estimation (QT_Blocking). The standard quantization table is used as a reference to find the closest quantization table for a JPEG fragment. The standard basis table $T_b[i]$ is scaled by

Table 2. Average values of coefficients.

519.39	17.37	9.27	5.79	2.59	0.82	0.32	0.12
22.59	11.36	7.35	4.84	1.62	0.44	0.20	0.12
12.61	8.32	5.91	2.66	0.86	0.32	0.16	0.10
8.04	5.91	2.91	1.56	0.46	0.13	0.07	0.05
5.39	2.70	1.18	0.49	0.23	0.05	0.02	0.02
2.23	0.99	0.47	0.22	0.08	0.03	0.01	0.01
0.55	0.26	0.15	0.09	0.03	0.01	0.01	0.01
0.15	0.09	0.07	0.04	0.02	0.01	0.01	0.01

a quality factor Q to obtain the new table $T_s[i]$. The scaling function is given by:

$$T_s[i] = \left\lfloor \frac{S \times T_b[i] + 50}{100} \right\rfloor \tag{6}$$

where $T_b[i]$ is the standard quantization table, Q is the quality factor that ranges from 1 to 100, $S = (Q < 50)?\frac{5000}{Q} : 200 - 2Q$, and if $T_s[i] \leq 0$, then $T_s[i] = 1$.

The advantage of the discrete cosine transform is its tendency to aggregate most of the signal in one corner of the result. In other words, most of the energy is concentrated in the low-frequency area located at the top-left corner. Due to the fact that the human eye is not sensitive enough to distinguish high-frequency brightness variations, when designing a quantization table, the quantization steps in the high-frequency area are usually larger than those in the low frequency area. Thus, the information of many zero-valued high-frequency components can be greatly reduced without much visual impact. Depending on the compression ratio, most AC coefficients have zero values after dividing the quantization steps – especially those at the high frequencies – in a zigzag pattern [5]. As a result, most of the energy in the original image is concentrated in the non-zero low frequency coefficients, whereas the DC coefficient is the most important coefficient.

Property 3. Quantized Zeros: *After discrete cosine transformation and quantization, the low-frequency coefficients are usually coded as a few non-zero coefficients in the top-left of image blocks and the high-frequency coefficients are coded as many zero-valued coefficients in the bottom-right.*

Experiments were conducted on 500 pictures taken by several popular digital cameras and smartphones. Table 2 shows the absolute mean values within a decoding block while Table 3 shows the probabilities of

Table 3. Probabilities of quantized zero coefficients (%).

0.04	6.46	11.11	16.78	33.11	58.86	77.92	91.60
5.38	9.38	13.20	18.50	41.61	69.29	85.69	91.25
10.38	13.16	17.48	37.46	61.56	76.74	88.89	92.37
15.56	18.58	37.01	44.06	74.53	88.70	94.88	95.73
24.82	38.92	57.31	73.52	83.75	95.37	98.15	97.96
46.08	61.50	74.06	85.32	94.18	97.97	99.24	99.06
70.94	82.00	87.45	93.70	97.83	99.27	99.56	99.38
89.11	91.95	93.55	96.15	98.43	99.09	99.36	99.18

the quantized zero coefficients. The results satisfy Property 3. The coefficients in the high-frequency area not only have smaller values, but also have higher probabilities of being zero. Therefore, when reconstructing a quantization table, the most important task is to correctly predict the quantization steps in the low-frequency areas.

In a JPEG image, the pixels of the original color YCbCr components in the spatial domain are coded using eight bits (corresponding to the range $[0, 255]$). After the discrete cosine transformation, the DC and AC coefficients take up eleven bits (corresponding to the range $[-1, 024, 1, 023]$). Conversely, in the decoding process, the coefficients must be dropped from the range $[-1, 024, 1, 023]$ after dequantization. Although the DC coefficients may be decoded incorrectly, the amplitudes of the DC coefficients are fixed. The amplitudes of the DC and AC coefficients can be found from the maximum and minimum coefficients in a decoding block. Therefore, the upper bound for the first quantization step can be restricted by dividing 2,048 by the amplitude of a DC coefficient as follows:

$$DC : Q_{ij} \leq \left\lfloor \frac{2,048}{abs(max(S_{ij})) + abs(min(S_{ij}))} \right\rfloor \tag{7}$$

where $(i, j) \in X$, $X = \{(i, j) | i \ mod \ 8 = 0 \wedge j \ mod \ 8 = 0\}$.

Equivalently, the upper bound of the quantization step of an AC coefficient is given by:

$$AC : Q_{ij} \leq \left\lfloor \frac{1,024}{max(abs(S_{ij}))} \right\rfloor \tag{8}$$

where $(i, j) \in \overline{X}$, S_{ij} is the decoded entropy coefficient at horizontal frequency i and vertical frequency j, and Q_{ij} is the estimated upper-bound quantization step for coefficient S_{ij}.

As shown in Table 2, the DC coefficient holds the most energy in each minimum coding unit, so the rough range of the estimated quantization

Table 4. Width estimations of two fragment groups.

Dataset	**Large Fragments (242)**		**Small Fragments (28)**	
Metric	**Accuracy(%)**	**Time(s)**	**Accuracy(%)**	**Time(s)**
Brute Force	98.35	183.6	75.00	1.094
Similarity Search	95.46	0.282	64.29	0.093

table can be improved by only finding the step at Q_{00}. Generally, Q_{00} can be restricted to a very small range such as less than two when the quality factor is larger than 92.2 and one when the quality factor is greater than 95.4. The Blocking_IQA algorithm is used in a brute force manner to evaluate pictures with all possible quality factors. The picture with the best quality score is the best estimated result.

5. Experimental Results

The performance of the proposed methodology was evaluated using an SD card containing approximately 1 GB of JPEG files – 207 high-resolution JPEG files and 120 unique quantization tables. All the pictures were captured by a BenQ DC E1050 digital camera. In the experiment, all the file headers were removed to simulate fragments without file headers. A total of 270 fragments were obtained, including sequential and fragmented files. Fragments that are too small have insufficient information for decoding. Therefore, the fragments were divided into two sets: 28 small fragments containing than 50 KB of data and 242 large fragments.

The computer used in the experiments had a 3.40 GHz CPU and 4 GB memory. Matlab (version 2013b) was used for all the computations, `libjpeg` (version 8b) was used as the default JPEG decoder/encoder and JPEG Toolbox (version 2.0) was used for coefficient extraction.

The proposed width estimation algorithm, called the similarity search algorithm, was applied to the small and large fragment sets. Table 4 shows the results along with those obtained using the brute force JPEG width estimation technique of Xu and Xu [18]. The similarity search algorithm yields an average accuracy of 95.46% for large fragments with an average time requirement of 0.282 seconds. Although the brute force algorithm has a better accuracy of 98.35%, the average time requirement per fragment is extremely high (183.6 seconds). Due to the limited information provided by small fragments, both the algorithms do not perform well. It is also pertinent to note that eight of the 28 small fragments had sizes less than 32 KB, meaning that the decoded pixel data was much less than the width of the image.

Since both the width estimation algorithms did not perform well on small fragments, additional quantization table prediction experiments were only conducted on the large fragment dataset. The quantization estimation experiments used the original picture and the saturated overflow algorithm of Huang et al. [4] for purposes of comparison.

The deviation of each element in the estimated quantization table was evaluated as:

$$\Delta Q = \frac{|Q_O - Q_E|}{|Q_O|} \tag{9}$$

where Q_O is the original quantization table and Q_E is the estimated quantization table.

The results show that the QT_Blocking algorithm performs better that the saturated overflow algorithm. The mean value of the deviation for the QT_Blocking algorithm was 0.1631 whereas the saturated overflow algorithm had a mean value of 0.6556.

Image quality was also evaluated using SSIM [15]. SSIM generates a score between zero and one, with a higher score indicating greater similarity between a test picture and original picture. A total of 157 beginning fragments were used for testing since the original fragments could be recovered easily from the image dump. The QT_Blocking algorithm had an average quality score of 0.9939 and required only 18.13 seconds; on the other hand, the saturated overflow algorithm had an average quality score of 0.9587 and required 142.5 seconds. Both algorithms had some quality scores lower than 0.85 due to failures in width prediction.

The performance of Blocking_IQA was also compared against that of JPEG_IQA, a popular no-reference image quality assessment algorithm [16]. After evaluating the rough quality factor range, the Blocking_IQA and JPEG_IQA algorithms were applied to find the picture with best quality score from among all the conditional quantization tables. The dataset comprised 207 complete pictures (the quantization tables were removed). Also, SSIM [15] was chosen as the reference algorithm to evaluate the image quality between the original picture and the estimated quantized files. The experiments revealed that the quality scores of pictures estimated by Blocking_IQA were higher than those of JPEG_IQA. Also, the Blocking_IQA required 0.7509 seconds for one file per round while JPEG_IQA required 1.3102 seconds. In other words, when assessing the quality of JPEG image files with incorrect quantization tables, the Blocking_IQA yielded better results than the well-known JPEG_IQA algorithm.

6. Conclusions

This methodology proposed in this chapter is specifically designed to recover partially-damaged standalone JPEG fragments by reconstructing pseudo headers. Extensive experiments with real camera pictures demonstrate that it can recover fragments with missing headers accurately and efficiently. However, the no-reference nature of the methodology renders it effective only on original pictures; this is because blocking features may be changed in edited or doubly-compressed pictures. Future research will examine blocking features in pictures in the context of image quality assessment as well as image forgery detection. Also, it will focus on extending the approach to other multimedia files such as videos.

Acknowledgement

This research was partially supported by International Cooperation and Exchange Project No. 61361166006, National Nature Science Foundation of China Project No. 61401176 and Natural Science Foundation of Guangdong Province (China) Project No. 2014A030310205.

References

[1] B. Carrier, *File System Forensic Analysis*, Pearson Education, Upper Saddle River, New Jersey, 2005.

[2] H. Farid, Digital image forensics, *Scientific American*, vol. 298(6), pp. 66–71, 2008.

[3] S. Garfinkel, Carving contiguous and fragmented files with fast object validation, *Digital Investigation*, vol. 4(S), pp S2–S12, 2007.

[4] L. Huang, M. Xu, H. Zhang, J. Xu and N. Zheng, A method of approximately reconstructing JPEG quantization table via saturated overflow, *Proceedings of the SPIE International Conference on Graphic and Image Processing*, vol. 8768, 2013.

[5] International Telecommunication Union, Information Technology – Digital Compression and Coding of Continuous-Tone Still Images – Requirements and Guidelines, Recommendation T.81, Geneva, Switzerland, 1993.

[6] M. Karresand and N. Shahmehri, Reassembly of fragmented JPEG images containing restart markers, *Proceedings of the European Conference on Computer Network Defense*, pp. 25–32, 2008.

[7] S. Klein and Y. Wiseman, Parallel Huffman decoding with applications to JPEG files, *The Computer Journal*, vol. 46(5), pp. 487–497, 2003.

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

[9] A. Mittal, A. Moorthy and A. Bovik, No-reference image quality assessment in the spatial domain, *IEEE Transactions on Image Processing*, vol. 21(12), pp. 4695–4708, 2012.

[10] A. Pal, H. Sencar and N. Memon, Detecting file fragmentation points using sequential hypothesis testing, *Digital Investigation*, vol. 5(S), pp. S2–S13, 2008.

[11] A. Pal, K. Shanmugasundaram and N. Memon, Automated reassembly of fragmented images, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 732-735, 2003.

[12] H. Sencar and N. Memon, Identification and recovery of JPEG files with missing fragments, *Digital Investigation*, vol. 6(S), pp. S88–S98, 2009.

[13] K. Shanmugasundaram and N. Memon, Automatic reassembly of document fragments via context based statistical models, *Proceedings of the Nineteenth Annual Computer Security Applications Conference*, pp. 152–159, 2003.

[14] G. Sharma and R. Bala (Eds.), *Digital Color Imaging Handbook*, CRC Press, Boca Raton, Florida, 2002.

[15] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Transactions on Image Processing*, vol. 13(4), pp. 600–612, 2004.

[16] Z. Wang, H. Sheikh and A. Bovik, No-reference perceptual quality assessment of JPEG compressed images, *Proceedings of the International Conference on Image Processing*, vol. 1, pp. I-477–I-480, 2002.

[17] M. Xu, L. Huang, H. Zhang, J. Xu and N. Zheng, Recovery method for JPEG file fragments with missing headers, *Journal of Image and Graphics*, vol. 18(1), pp. 24–35, 2013.

[18] Y. Xu and M. Xu, Width extraction of JPEG fragment via frequency coefficients scale similarity measuring, *Proceedings of the Second International Conference on Future Computer and Communication*, vol. 2, pp. 513–517, 2010.