JPEG IMAGE SCRAMBLING WITHOUT EXPANSION IN BITSTREAM SIZE

Kazuki Minemura^a*, Zahra Moayed*^{a}*, KokSheik Wong*^{a}*, Xiaojun Qi*^{b} *and Kiyoshi Tanaka*^{c}

^{a}{kazuki.minemura, moayed135}@gmail.com, koksheik@um.edu.my
Faculty of Comp. Sci. & Info. Tech, University of Malaya, Kuala Lumpur 50603, Malaysia.
^{b}Xiaojun.Qi@usu.edu
Department of Computer Science, Utah State University, Logan, UT 84322-4205, USA.
^{c}ktanaka@shinshu-u.ac.jp
Faculty of Engineering, Shinshu University, Nagano 380-8553, Japan.

ABSTRACT

In this work, an algorithm is proposed to scramble an JPEG compressed image without causing bitstream size expansion. The causes of bitstream size expansion in the existing scrambling methods are first identified. Three recommendations on AC coefficients in the scrambled image are proposed to combat unauthorized viewing. As the first step of the scrambling algorithm, edges are identified directly in the frequency domain using solely AC coefficients without relying on any traditional methods. These edges then form a low resolution image of its original counterpart and the information is utilized to identify regions. The DC coefficients are encoded in region-basis to suppress bitstream size expansion while achieving scrambling effect. Experiments were carried out to verify the basic performance of the proposed scrambling method. For the parameter settings considered, most of the scrambled images are of smaller bitstream size than their original counter parts.

Index Terms— Nonzero coefficient count, scrambling, DCT, JPEG, edge detection

1. INTRODUCTION

JPEG compression standard is proposed since 1992 [1] but yet it still proves its importance in our daily applications thanks to the vast number of decoders that support its format, commercial digital cameras that capture in JPEG mode, and the existing images stored in this format. For that, various fields of study including scrambling, data embedding, image enhancement, etc. specifically designed for JPEG compressed images are still receiving much attention in the research community [2, 3, 4].

Among these topics, scrambling algorithms are of high popularity because they provide privacy to the captured JPEG images. Some of the notable approaches are sign randomization [5], permutation of zero-run level pairs within a block [6], permutation of coefficients within sub-bands [7], and global permutation of zero-run level pairs [4]. While [5] maintains the original bitstream size, the original image can be easily sketched from the scrambled counterpart as detailed in [7]. On the other hand, although [4, 6] and [7] achieve certain level of robustness against unauthorized viewing, they suffer from bitstream size expansion. These expansions are due mainly to (i) the destruction of the correlation between the number of zero and nonzero coefficients [7], (ii) randomization of DC coefficients [6], or (iii) introduction of foreign nonzero coefficients [8], etc.

Some effort was carried out recently in [8] to suppress bitstream size expansion due to scrambling. In particular, 7 static and 1 image dependent scanning orders for AC coefficients are considered to reduce the bitstream size of the original image. However, this gain is not able to offset the side effect of bitstream size expansion due to the scrambling of DC coefficients.

In this work, an algorithm is proposed to scramble an JPEG compressed image without causing bandwidth expansion. Three recommendations on AC coefficients in the scrambled image are proposed to tackle the problem of unauthorized viewing. Before the first stage of the proposed scrambling algorithm, edge features are detected directly in the frequency domain using solely AC coefficients. These edges then form a low resolution image of its original counterpart and guide the region identification process. To suppress bitstream size expansion while achieving scrambling effect, the DC coefficients are encoded in region-basis. Experiments were carried out to verify the basic performance of the proposed method.

2. JPEG COMPRESSION

A grayscale image is divided into non-overlapping blocks of size 8×8 pixels. Each block is then transformed into DCT (Discrete Cosine Transform) domain [1]. The DCT coefficients are then quantized using the divisors $QT[u][v]$ determined by the quality factor. The quantized AC coefficients undergo zigzag ordering and entropy encoding. At the same time, the previous quantized DC coefficient is used to predict the current quantized DC coefficient. The prediction errors of all quantized DC coefficients are then entropy coded. For the rest of the paper, we denote $G(i, j)$ as the (i, j)-th 8×8 quantized coefficient block in a JPEG image, and denote $G_{u,v}(i, j)$ as the (u, v)-th element in $G(i, j)$ for $1 \leq u, v \leq 8$. During reconstruction (i.e., decoding), AC coefficients are reconstructed using:

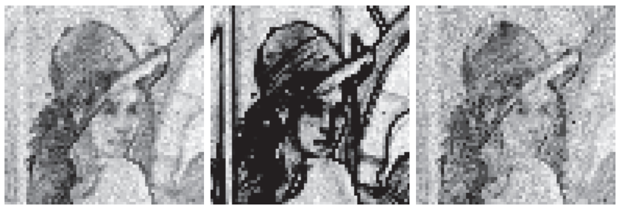

(a) NCC (b) EAC (c) PLZ

Fig. 1. Edge image sketched by using NCC, EAC and PLZ

$$rec[u][v] = C \times G_{u,v}(i,j) \times QT[u][v], \tag{1}$$

where C is a constant. The rest of the decoding processes is the opposite of the encoding phase.

3. EDGE INFORMATION FROM AC COEFFICIENTS AND THREE RECOMMENDATIONS

It is a known fact in DCT that the DC coefficient represents the average intensive value of the 8×8 block, and AC coefficients provide the detailed information about the block [9]. However, Li and Yan [7] suggested that the number of nonzero AC coefficients in a block can also be exploited to identify characteristics of the block, and hence the image. Generally, blocks with more nonzero coefficients contain edge/texture information, while low count in nonzero coefficients infers a smooth area. Following this fact, they proposed a method called nonzero count attack (NZA) to sketch outline of a scrambled JPEG image using only its AC coefficients.

To this end, we propose three simple methods by improving the idea in [7] to produce edge information of a JPEG compressed image using solely its AC coefficients. In particular, NCC (nonzero coefficients count), EAC (energy of AC coefficients), and PLZ (position of last nonzero coefficients) are defined as follows to generate the outline (edge image) f:

NCC: Nonzero coefficient count

$$f_1(i,j) \leftarrow \text{round}\left(255 \times \frac{c(i,j)}{\max_{(i,j)}\{c(i,j)\}}\right) \tag{2}$$

where $c(i,j)$ denotes the number of nonzero coefficients in $G(i,j)$;

EAC: Energy of AC coefficients

$$f_2(i,j) \leftarrow \frac{[\sum_{u=1}^{8}\sum_{v=1}^{8}|G_{u,v}(i,j)|] - |G_{1,1}(i,j)|}{\bar{f}_2} \tag{3}$$

where $|G_{u,v}(i,j)|$ gives the magnitude of $G_{u,v}(i,j)$ and

$$\bar{f}_2 \leftarrow \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{[\sum_{u=1}^{8}\sum_{v=1}^{8}|G_{u,v}(i,j)|] - |G_{1,1}(i,j)|}{M \times N} \tag{4}$$

PLZ: Position of last nonzero AC coefficients

$$f_3(i,j) \leftarrow \frac{G_{u_0,v_0}(i,j)}{\bar{f}_3} \tag{5}$$

where $G_{u_0,v_0}(i,j)$ denotes the position of the last nonzero coefficient in $G(i,j)$ with respect to the zigzag scanning order and $\bar{f}_3 \leftarrow \max_{(i,j)}\{G_{u_0,v_0}(i,j)\}$.

Fig. 1 shows the sketch outline produced by each of the three methods. Although the size of the output image is 8×8 smaller than its original counterpart, the AC coefficients reveal edge information of the original image. Compared to the method [7] that generates a binary output image, the proposed method does not require the tuning of threshold parameters and its output image is of higher fidelity. Nevertheless, result similar to that of [7] can be produced by thresholding (be it a fixed value or adaptive) the output of NCC, EAC, or PLZ. Note that the aforementioned ideas are applicable to any block-transform coding approach, which will be further explored as our future work.

The direct implication of these findings to the design of the scrambling algorithm in the DCT compressed domain is that NCC, EAC, and PLZ must be modified for each block to survive the above attacks. Otherwise, the 8×8 coefficient blocks must be shuffled (like jigsaw puzzle) to battle against unauthorized viewing or leakage. However, it is a known fact that scrambling DC coefficients will lead to bitstream size expansion due to inefficiency of DPCM (differential pulse coding modulation) [1]. Scrambling only the AC coefficients (intra sub-band or block permutation) while keeping the DC coefficients intact is not feasible because the unprocessed DC coefficients (representing the average pixel intensity value of a block) will immediately reveal the outline of the image.

4. REDUCING BITSTREAM SIZE EXPANSION IN JPEG COMPRESSED IMAGE

DPCM and the entropy encoding of DC coefficients using Huffman codewords exploit the small variation of values[1] among adjacent blocks to achieve compression. As a result, we aim to arrange the DC coefficients so they are in certain order. In the ideal case with respect to the codeword length, DC coefficients can be coded using the least number of bits when they are arranged in monotonic non-decreasing (or non-increasing) order. However, the overhead needed to keep track of the original location of each DC coefficient is more than the compression gained.

In our proposed algorithm, we opt for the sub-optimal approach since pixel intensity values within an object or a connected region are of subtle variation. In particular, we group 8×8 coefficient blocks based on the edge image generated in **Section 3**. To fasciliate discussion, we consider the edge image generated by EAC since it gives the best visual result.

The flow of operations is illustrated in **Fig. 2**. The output of EAC is first binarized by using Otsu's approach [10], and connected regions in the binary image are labelled using the 4-connectivity [11]. DC coefficients originating from blocks of the same label will be encoded using the existing technologies in JPEG (i.e., DPCM + Huffman coding). An example of the processed image is as shown in **Fig. 3(a)**. It is observed that slices of blocks, each of almost the same intensity value, are formed (particularly at the lower part of the image).

[1]Each pixel represent the average pixel intensity value in a block

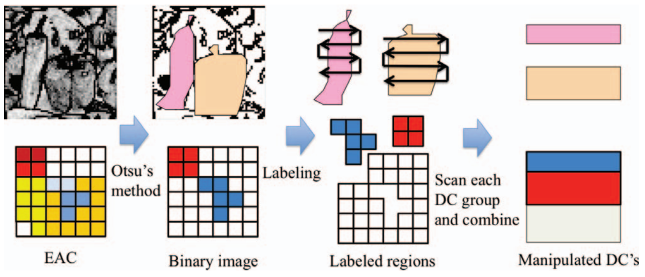


Fig. 2. Flow of operations in encoding DC coefficients

This implies that DC coefficients of approximately the same value are grouped together. It should be noted that the similar results can be obtained when NCC or PLZ sketch outline is utilized. Similarly, the algorithm considered for each operation (e.g., labeling) can be replaced by any other algorithm achieving the same objective(s), and the results are expected to be almost the same. The important point to note here is that, blocks belonging to an object or connected region will have small variation in pixel intensity value. This in turn implies that the DC value of these blocks are also of small variation.

5. PROPOSED SCRAMBLING METHOD

The processing of DC coefficients provides some form of scrambling effect (because neither the original image nor its outline counterpart is readily decodable), but edges in the original image are still visible as suggested by **Fig. 3(a)**. In this section, we propose a complete scrambling algorithm that does not cause bitstream size expansion. First, the sign of each nonzero AC coefficient is XOR-ed with a pseudo-random sequence of -1 and 1's. Secondly, for each 8×8 block, zerorun-level pairs are formed and permuted within the block. Third, the modified blocks are then permuted globally (e.g., the first block $G''(1, 1)$ becomes $G''(3, 2)$). Note that these three algorithms will not affect the bitstream size, except for a small variation due to byte-alignment in JPEG [1]. Finally, DC coefficients are processed in groups as described in the previous section. If the proposed method handling DC coefficient does not affect the bitstream size, the proposed scrambling method will maintain the bitstream size even when the image is scrambled. The processes are completely reversible where the original image can be regenerated, and its presentation is omitted here.

The robustness of this method against unauthorized viewing is achieved by consolidating scrambling algorithms designed to manipulate specific entities in the JPEG compression standard. For brute force attack, the attacker needs to first guess the correct order for block shuffling ($(M \times N)!$ trials for an image of dimension $8M \times 8N$ pixels), and for each block, the original order of zerorun-level pairs must also be guessed correctly (maximum 63! trials within each block). Finally, the sequence of -1 and 1's to XOR with the sign of nonzero coefficients must also be guessed correctly (maximum $2^{8M \times 8N}$ trials). On top of that, the DC coefficients are also processed. Therefore, we conclude that our method is relatively robust against unauthorized viewing or leakage, at least from the brute force attack point of view.

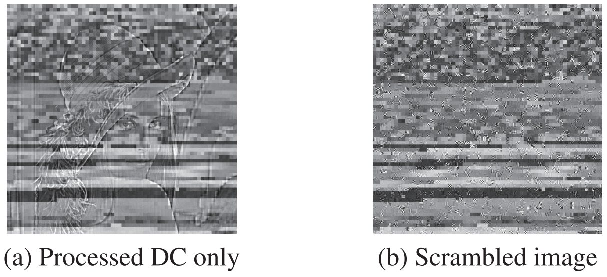


Fig. 3. Example of processed image

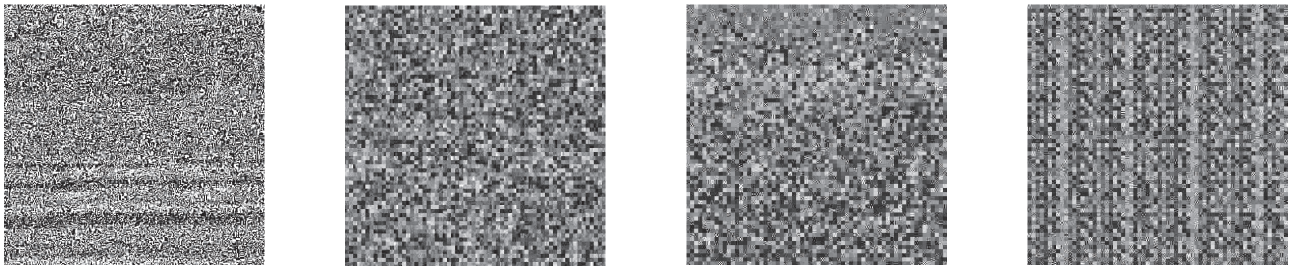
An example of the scrambled image (Lenna) is shown in **Fig. 3(b)**. As expected, the image appears to be scrambled and the outline of Lenna is no longer visible. Note that the distortion level of the scrambled image can be further intensified (say to noise-like image as in [7]) by changing the value of the quantization table in use. In particular, if value of the divisors $QT[u][v]$ are forced to any large value, the reconstructed coefficients will be of large magnitude according Eq. (1). Pixel intensity values of the corresponding block will be saturated to 255 or 0 for an 8-bit image.

6. EXPERIMENTAL RESULTS AND DISCUSSIONS

Six standard test images (i.e., Airplane, Baboon, Boat, Lake, Lenna and Peppers) are considered to verify the basic performance of the proposed scrambling method. The quality factor is set to 75. "Mersenne Twister"[12] with 32-bit seed is utilized as the pseudo-random number generator. It is verified that the scrambled image can be de-scrambled to obtain the original image.

We consider the bitstream size expansion of the proposed method. For comparison purposes, we also considered the results for FIBS [7], Takayama et al.'s approach [6] and Wong et al.'s method [8]. The results are summarized in **Table 1**. The results indicate that the proposed scrambling method produces image of smaller bitstream size than its original counter part or of insignificant expansion $<0.1\%$, while the existing methods always produce image of larger bitstream size. The comparison to [6] is particularly important because it clearly indicates that a simple permutation of DC coefficients will lead to bitstream size expansion while the proposed preprocessing of DC coefficients in **Section 4** suppresses bitstream size expansion. The expansion for FIBS [7] is particularly high because coefficients are scrambled within subbands in which case the statistics of zerorun-level pairs for AC coefficients and correlation of DC coefficients are completely destroyed.

For completion of discussion, **Fig. 4** shows the scrambled image of Lenna for each method considered. Here, the quantization table in the image generated by the proposed method is modified (i.e., all divisors are set to 255, except for the DC component). It can be seen that each method successfully distorts the perceptual quality of the input image. Some repeating patterns are observed for Wong et al.'s method [8]

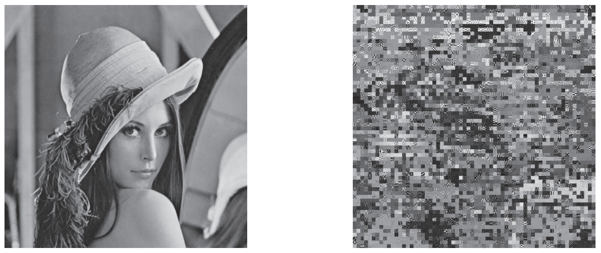


(a) Proposed + Change QT (b) FIBS[7] (c) Takayama [6] (d) Wong [8]

Fig. 4. Scrambled image for various methods

Table 1. Percentage of bitstream size expansion [%]

Image	Proposed	Ref [7]	Ref [6]	Ref [8]
Airplane	-0.037	25.678	3.945	11.187
Baboon	0.039	10.655	1.433	4.396
Boat	0.094	17.223	2.467	8.573
Lake	-0.194	16.656	3.428	8.921
Lenna	-0.335	21.439	4.150	11.623
Peppers	-0.076	17.673	4.472	11.688
Average	-0.085	18.221	3.315	9.398



(a) Same key (b) Wrong key (differs by 1 bit)

Fig. 5. Example of de-scrambled image using different keys

while the 'stripes' that appear in **Fig. 3(a)** and **(b)** are still subtly visible for the proposed method. Nevertheless, the proposed method clearly shows its superiority in terms of bitstream size expansion. In fact, some compression is actually gained. **Fig. 5** shows two de-scrambled images reconstructed by using two different keys: the correct seed and the incorrect seed. It clearly shows that a wrong key will fail to regenerate the original image. In particular, the perceptual similarity is very low even when the seed differs only by one bit.

7. CONCLUSIONS

A novel algorithm was proposed to scramble JPEG compressed image without bitstream size expansion. Regions were formed based on the edge information induced by the AC coefficients using any of the three recommendations, and DC coefficients from the same region are encoded in group. Zerorun-level pairs within a block were shuffled, followed by the permutation of coefficient blocks. It was verified that the proposed method could distort the visual quality of the input image while not leaking the sketch outline of the original image. The proposed method was also compared to recent methods, and results suggested that the proposed method causes significantly smaller bitstream size increment than the existing methods considered, with occasional smaller bitstream size than the original image.

As future work, we want to further suppress the bitstream size expansion and exploit the information induced by AC coefficients in applications such as image processing and compression.

8. REFERENCES

[1] William B.Pennebaker and Joan L.Mitchell, *JPEG: still image data compression standard*, Van Nostrand Reinhold, 1992.

[2] G. A. Triantafyllidis, M. Varnuska, D. Sampsona, D. Tzovaras, and M. G. Strintzis, "An efficient algorithm for the enhancement of JPEG coded images," *Computers & Graphics: An International Journal of Systems & Applications in Computer Graphics*, vol. 27, pp. 529–534, 2003.

[3] M. Fujiyoshi, K. Kuroiwa, and H. Kiya, "A scrambling method for motion JPEG videos enabling moving objects detection from scrambled videos," *In Proc. IEEE ICIP*, pp. 773–776, 2008.

[4] K. Wong and K Tanaka, "DCT based scalable scrambling method with reversible data hiding functionality," *In Proc. IEEE ISCCSP*, pp. 1–4, 2010.

[5] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," *IEEE Trans. Circuits Syst. Video Technol*, vol. 17, pp. 774–778, 2007.

[6] M. Takayama, K. Tanaka, A. Yoneyama, and Y. Nakajima, "A video scrambling scheme applicable to local region without data expansion," *In Proc. IEEE ICME*, pp. 1349–1352, 2006.

[7] W. Li and Y. Yuan, "A leak and its remedy in JPEG image encryption," *Int. J. Comput. Math*, pp. 1367–1378, 2007.

[8] K. Wong, S. Ong, K. Tanaka, and X. Qi, "Bitstream size suppression for DCT-based information hiding method," *In Proc. IEEE ISPACS*, pp. 7–9, 2011.

[9] K.R.Rao and P.Yip, *Discrete cosine transform: algorithms, advantages, applications*, Academic Press, 1990.

[10] N.Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems*, vol. 9, no. 1, pp. 62–66, 1979.

[11] Haralick Robert M and Linda D. Shapiro, *Computer and robot vision*, vol. 1, Addison-Wesley, 1992.

[12] M. Matsumoto and T. Nishimura, "Mersenne twister :a 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Trans. on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, 1998.