PROGRESSIVE QUALITY DEGRADATION IN JPEG COMPRESSED IMAGE USING DC BLOCK ORIENTATION WITH REWRITABLE DATA EMBEDDING FUNCTIONALITY

SimYing Ong, Kazuki Minemura, KokSheik Wong

simying@siswa.um.edu.my, kazuki.minemura@gmail.com, koksheik@um.edu.my
Faculty of Comp. Sci. & Info. Tech., University of Malaya, 50603 Kuala Lumpur, Malaysia.

ABSTRACT
This paper proposes a novel block rotational method to degrade quality and embed external data in JPEG compressed image. The orientation of each non-overlapping DC coefficients block is exploited to embed information while introducing distortion. To achieve progressive quality degradation, size of DC coefficients block is manipulated and the proposed embedding process is applied recursively by shrinking block size in each iteration. Markers are added into the blocks as pre-processing steps to ensure that the original orientation always yields the smallest difference. A post-processing is also proposed to erase the marker introduced for recovering image at higher quality, making the proposed method a rewritable method but not complete reversible. Experiments are conducted to verify the basic performance of the proposed method and comparisons with the conventional methods are also carried out.

Index Terms— Block orientation, progressive quality degradation, rewritable data embedding, JPEG

1. INTRODUCTION

With the ever improving capabilities of affordable smart devices and ubiquitous network environment, image can be easily captured, edited, and broadcasted through social networking site or stored remotely in cloud. JPEG is arguably the most famous image compression standard due to its relatively simple architecture and good trade off between bitrate (i.e, size) and quality [1]. For that, various researches are carried out to manage, protect, or enhance JPEG compressed image (hereinafter referred to as image). In particular, quality degradation techniques were proposed by manipulating the underlying coding structure of JPEG bitstream to protect the image from unauthorized viewing. Data embedding techniques are proposed to encode information, including hyperlink to related contents, hash / check-sum value for integrity checking, watermark for claiming ownership, etc.

For quality degradation, most of the techniques rely on the manipulation of DC and AC coefficients stored in the JPEG bitstream [2, 3]. Some researchers focus on randomizing the sign of transformed coefficients [4, 5] and shuffling the AC coefficients within the block (i.e., intra-block) [6] or within the image (i.e., inter-block) [3, 7]. However, some methods [3, 6] cause significant bitstream size increment because the zero-run-value structure of AC coefficients is modified. Furthermore, [4, 5, 6] is vulnerable to the sketch attack where low resolution image can be revealed in the attacks proposed in [3, 7]. On the other hand, DC coefficients must also be handled because DC coefficients themselves can produce a sketch of the JPEG image, but at one-eighth of the original resolution. Therefore, several methods are proposed to manipulate DC coefficients while aiming to minimize the bitstream size overhead. Zeng et al. propose to randomize the sign of DC coefficients [4] while Niu et al. propose to replace the original prediction error of DC coefficient by another error belonging to the same category [8]. Minemura et al. partition a JPEG image into regions based on the information obtained from AC coefficients and DC coefficients are handled independently on regional basis [2].

Recently, data embedding and quality degradation are combined for content management purposes [7, 9, 10]. For example, in the cloud storage environment, image can be severely distorted to protect privacy of the owner and metadata describing the image can be embedded. The administrator can extract the embedded metadata to further process the image (e.g., move, copy, delete, etc.) without the need to know the actual visual appearance of the image.

In this work, a progressive quality degradation method equipped with rewritable data embedding capability is proposed. Both quality degradation and data embedding are achieved by exploiting the orientation of DC coefficients grouped in blocks of pre-defined sizes. The level of distortion and embedding capacity are controlled by the block size and the number of times the proposed method is applied recursively. Basic performance of the proposed method are verified through experiments using standard text images.

2. DC BLOCK ROTATIONAL METHOD

The array of DC coefficients $D$ is first constructed by collecting the DC component from each $8 \times 8$ block. $D$ is then divided into non-overlapping blocks $B(i, j)$ each of size $b \times b$ coefficients for $1 \leq i \leq |M/b|$ and $1 \leq j \leq |N/b|$ where $8M \times 8N$ is the dimension of the input image. Each block is rotated to host external data by exploiting the relative orien-
let $\varepsilon_k(i, j)$ denote the error associated with $R_k$ for $k = 1, 2, 3$ and 4. The objective here is to ensure that $R_3$ yields the smallest error. The errors $\varepsilon_k$'s are first sorted in increasing order. If $R_3$ yields the smallest error, then nothing is done. Otherwise the orientation $R_{k_0}$ which yields a smaller error than $R_3$ is determined. The location $(x, y)$ along the first column and first row of the block in question (after the rotating $R_1$ by $(k_0 - 1) \times 90^\circ$) are searched to find for the smallest difference $\delta_{k_0}(x', y')$ using:

$$
(W + S + E)/3 - X
$$

for $x' = 1$ and $2 \leq y'b \leq b - 1$ (i.e., first row), or

$$
(N + E + S)/3 - X
$$

for $2 \leq x'b \leq b - 1$ and $y'b = 1$ (i.e., first column). Here, $X = DC(x, y), N = DC(x - 1, y), E = DC(x, y + 1), W = DC(x, y - 1), S = DC(x + 1, y),$ and $\%$ denotes the modulus operation. The value, i.e., $DC(x', y')$, is replaced by a marker (i.e., a big value). In this work, the marker value is

$$
\text{sgn}(DC(x', y')) \times (\lfloor DC_{\text{max}} \rfloor + 1),
$$

where $\text{sgn}(DC(x', y'))$ denotes the sign (i.e., ‘+’ or ‘−’) of $DC(x', y')$ and $\lfloor DC_{\text{max}} \rfloor$ is the largest DC coefficient in $D$ in terms of magnitude. This process increases $\varepsilon_{k_0}$ and leads to $\varepsilon_1 < \varepsilon_{k_0}$. The process is repeated until $R_1$ yields the smallest error. Note that the value $DC_{\text{max}}$ is required for recovery purposes and hence it must be sent to the receiver.

At the receiver’s end, since $R_1$ is pre-processed to yield the smallest error, the embedded information can be extracted by comparing the current orientation and that of $R_1$. To recover the input image, all blocks $B(i, j)$ are reset to their original orientations by considering $\varepsilon(i, j)$. The marker can be easily identified since its magnitude is larger than $DC_{\text{max}}$. In particular, if $|DC(x, y)| \leq |DC_{\text{max}}|$ holds true for all DC coefficients in $B(i, j)$, then $B(i, j)$ did not undergo the pre-processing, and a perfect reconstruction can be achieved. Otherwise, the marker is replaced by an estimated value. In particular, the estimation depends on the position at which the marker is located and it is detailed as follows:

$$
\text{DC}(i', j') = \begin{cases} 
(W + S + E)/3 & \text{top row}, \\
(N + E + S)/3 & \text{left column}, \\
(W + N + E)/3 & \text{bottom row}, \\
(N + W + S)/3 & \text{right column}.
\end{cases}
$$

It is expected that the estimated DC value (i.e., after post-processing) will be similar to that of the original one (i.e., before pre-processing) due to the correlation among DC coefficients. Thus, the recovered image after post-processing is expected to exhibit comparable quality w.r.t. the original image. Here, quality of the recovered image $A_1$ is lower than the original image $A^0$. However, $Q(A^n) = Q(A^1)$ holds true for $n \geq 1$ where $Q(\cdot)$ is an image quality metric and $A^n$ denote the image reconstructed from the scrambled-embedded image when $A^{n-1}$ is utilized as the input. Therefore, the proposed method achieves the rewritable property.

![Fig. 1. Four possible orientations for DC block $B(i, j)$](image-url)}
### Table 1. SSIM and bitstream size for 8 x 8 block

<table>
<thead>
<tr>
<th>Test Image</th>
<th>SSIM</th>
<th>Original (Basic)</th>
<th>Output (Enh.)</th>
<th>Recovered (Basic)</th>
<th>Recovered (Enh.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td></td>
<td>0.966</td>
<td>0.194</td>
<td>0.189</td>
<td>0.966</td>
</tr>
<tr>
<td>Baboon</td>
<td></td>
<td>0.943</td>
<td>0.028</td>
<td>0.026</td>
<td>0.943</td>
</tr>
<tr>
<td>Lake</td>
<td></td>
<td>0.921</td>
<td>0.063</td>
<td>0.060</td>
<td>0.921</td>
</tr>
<tr>
<td>Lenna</td>
<td></td>
<td>0.948</td>
<td>0.092</td>
<td>0.080</td>
<td>0.948</td>
</tr>
<tr>
<td>Peppers</td>
<td></td>
<td>0.914</td>
<td>0.068</td>
<td>0.070</td>
<td>0.913</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Image</th>
<th>Output Bitstream Size [KBytes]</th>
<th>Original (Basic)</th>
<th>Output (Enh.)</th>
<th>Recovered (Basic)</th>
<th>Recovered (Enh.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>37.82</td>
<td>39.68</td>
<td>39.84</td>
<td>37.83</td>
<td>37.85</td>
</tr>
<tr>
<td>Baboon</td>
<td>76.84</td>
<td>77.05</td>
<td>77.18</td>
<td>76.85</td>
<td>76.83</td>
</tr>
<tr>
<td>Lake</td>
<td>51.00</td>
<td>52.92</td>
<td>53.01</td>
<td>50.97</td>
<td>51.01</td>
</tr>
<tr>
<td>Lenna</td>
<td>37.05</td>
<td>38.90</td>
<td>39.07</td>
<td>37.05</td>
<td>37.04</td>
</tr>
<tr>
<td>Peppers</td>
<td>38.50</td>
<td>40.35</td>
<td>40.63</td>
<td>38.52</td>
<td>38.51</td>
</tr>
</tbody>
</table>

### 3. ENHANCEMENTS

This section improves the robustness of the proposed method against unauthorized viewing of the manipulated image. First, as mentioned in Section 2, the original orientation $R_1$ can be determined by considering $\varepsilon_k(i, j)$. Therefore, anyone can decode the embedded data and recover the image (i.e., plaintext) using the inverse process.

For that, some randomness is integrated into the preprocessing steps so that $R_1$ does not always yield the smallest error. In particular, a sequence of random integers $\{k_l\}_{l=1} \subset [1, 4]$ is generated and $B(i_l, j_l)$ will be processed so that the original orientation $R_1$ becomes the $k_l$-th smallest error. In other words, the random sequence will determine the order (w.r.t. difference) to be assumed by $R_1$ by modifying DC coefficients at the northern and western borders to accommodate the condition (i.e., order) imposed. For instance, if $k_1 = 2$ but the original orientation (i.e., $R_1$) yields the smallest error, then either border I and/or II (shown in Fig. 1) of $B(i_l, j_l)$ will be modified by using the marker to increase the error $\varepsilon_1(i_l, j_l)$. The borders will be modified until the condition is met.

Secondly, in JPEG standard, the DC coefficients are encoded using DPCM (differential pulse coding modulation) where only the prediction errors are stored. However, prediction errors in the same category are represented by Huffman codewords of the same length (i.e., bits). Therefore in this work, the original prediction error $\Delta(x, y)$ is bijectively mapped to another prediction error value $\Delta'(x, y)$ from the same category as detailed in [8]. Note that mapping will not affect the bitstream size while further degrading the image quality. However, we shall show in Section 4 that scrambled output attained by bijectively mapping the prediction errors themselves is insufficient to withstand the sketch attack.

Finally, AC coefficients are manipulated to further degrade quality of the image. Researchers often concentrate on one type of coefficient and neglect the other. Since both DC and AC coefficients are able to reveal sketch of the input image [2], both should be manipulated to avoid unauthorized viewing. To this end, ZRV (zororan value) pairs in the image are gathered (from the entire image or blocks of sizes $s \times s$ pixels), scrambled, and output. Here, the number of nonzero coefficients in each block is maintained, but the members of ZRV pairs are different. Since a sketch can be obtained through counting the nonzero coefficients [2], all blocks must be shuffled similar to jigsaw puzzle.

### 4. EXPERIMENTAL RESULTS AND DISCUSSIONS

Five standard test images (i.e., Airplane, Baboon, Lake, Lenna and Peppers each of dimension 512 x 512 pixels) are considered for experiment purposes. It is verified that the proposed method can achieve both quality degradation (by visual inspection) and data embedding. Embeded data can be completely extracted and the image can be recovered. For the rest the discussion, unless specified otherwise, the quality factor is set to 80 and $b = 8$.

First, quality of the output image is measured by using SSIM [11] and the results are recorded in the left column of Table 1. $Enh.$ refers to the output produced by consolidating the proposed methods detailed in Section 2 and 3, and $Basic$ is exactly the same as $Enh.$, except for the randomness introduced to combat unauthorized viewing. Here, the results are collected by applying $Basic$ and $Enh.$ once in non-overlapping blocks each of size $8 \times 8$ DC coefficients. Results suggest that quality of each image is severely distorted (since the SSIM values are far from unity). $Enh.$ causes more distortions because more markers were inserted into the image. For the recovered image, its quality is observed to be lower than that of the original JPEG for both $Basic$ and $Enh.$. This expected insignificant drop in SSIM ($\sim 2 \times 10^{-4}$ and $\sim 1.2 \times 10^{-4}$ for $Basic$ and $Enh.$, respectively) is due to the insertion of markers. Second column of Table 2 records quality of the output image when the proposed method is applied once to various block sizes and when it is applied iteratively to smaller blocks (from outer to inner centre). Results suggest that different level of distortion can be achieved by considering combination of different block size $b$ and different number of recursive layers. For completion of discussion, Fig. 2(a)-(e) show the output image for various combinations of the proposed quality degradation method and Fig. 2(f) shows the recovered image.

Secondly, bitstream size overhead (i.e., increment) is verified. Right column of Table 1 records the bitstream size of the original, degraded and recovered images. It is observed that the average overhead in the degraded image are $\sim 3.41\%$ and $\sim 3.76\%$ for $Basic$ and $Enh.$, respectively. The increase in bitstream size is due to the increment of prediction error (between DC coefficients from adjacent block) caused by the proposed block rotational degradation method, which leads to a slight drop of efficiency in DPCM coding. Bitstream size of the recovered images are also larger than their original coun-
terparts, but at insignificant magnitude of < 0.01%.

Thirdly, carrier capacity of the proposed method is recorded in the third column of Table 2 for various block sizes and iterative layers. While the capacity is relatively low, it should be noted that: (a) all of the conventional scrambling methods considered are not able to embed any external data, and; (b) AC coefficients, which are merely shuffled to degrade quality in this work, can be further exploited for high capacity reversible data embedding purposes.

Last but not least, the proposed method is compared to the DC based quality degradation method proposed by Niu et al. [8]. To this end, a simple sketch attack is proposed by assigning a representation values (e.g., 2c for category c) to each category of the error value. Fig. 3(a)-(c) shows the results of applying the proposed simple sketch attack on output generated by Niu et al.’s method [8], the proposed method applied once (single), and the proposed method applied recursively. By visual inspection, outline of the image scrambled by Niu et al.’s method can be seen while the proposed simple sketch attack fails to reveal any feature of the image generated by the proposed methods. Thus, we conclude that proposed method is more secure than [8] w.r.t. the proposed simple sketch attack. Next, the comparison in terms bitstream size overhead is also considered. Table 3 shows the bitstream size comparison with the related works in achieving quality degradation using

<table>
<thead>
<tr>
<th>Related Methods</th>
<th>Bitstream [KBytes]</th>
<th>Increment [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Image</td>
<td>38.50</td>
<td>-</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>40.20</td>
<td>4.41</td>
</tr>
<tr>
<td>Intra-Block Shuffle</td>
<td>71.22</td>
<td>84.98</td>
</tr>
<tr>
<td>Intra-Block Shuffle (AC Only)</td>
<td>65.14</td>
<td>69.21</td>
</tr>
<tr>
<td>DC Splitting ⊕ Intra-Block Shuffle [6]</td>
<td>71.19</td>
<td>84.91</td>
</tr>
<tr>
<td>Frequency Band Shuffle [3]</td>
<td>44.70</td>
<td>16.10</td>
</tr>
<tr>
<td>Frequency Band Shuffle (AC Only) [3]</td>
<td>43.05</td>
<td>11.82</td>
</tr>
</tbody>
</table>

Peppers as the representative image. Results indicate that bitstream size overhead is inevitable for most of the methods considered. Although Niu et al. [8] is able to preserve the size, it is not completely secure w.r.t. the proposed simple sketch attack. Minemura et al.’s [2] method also preserves the size (and occasionally gains further compression), it is not able to achieve scalable quality degradation and data embedding. Similar results are observed for other test images. For that, we conclude that the proposed method achieves higher robustness against sketch attacks and offers more functionalities at the expense of negligible bitstream size increment.

5. CONCLUSION

This paper proposed a novel DC block rotational method to simultaneously degrade image quality and embed external data by exploiting orientation information. Marker was added into the DC coefficients block during pre-processing to ensure that all four block orientations are completely distinguishable during the data extraction and recovery processes. AC coefficients were also permuted to further intensify the distortion. Post-processing was carried out after data extraction to improve image quality of the recovered image. Experiments verified that the proposed method is able to progressively degrade image quality by considering different combination of block size and number of recursive layer, and achieves rewritable data embedding. It outperforms the existing methods in terms of robustness against sketch attacks and offers more features.

As future work, we want to increase the carrier capacity by exploring AC coefficients and achieve greater flexibility in quality degradation. We also want to apply the proposed method in formats such as JPEG-XR and MPEG/H.264.
6. REFERENCES


