Abstract
Since digital contents are easy to copy and manipulate without the loss of quality, many social issues related to forgery and illegal distribution are raised to protect the copyright of contents. Reversible data hiding is an effective way which keeps the quality of the contents and embeds copyright information into contents itself. Although compression is inevitable for multimedia, few reversible data hiding researches against compression have been performed. In this paper, we propose a novel reversible data hiding algorithm that is applied to authenticate commonly used JPEG images. The differential histogram of the quantized coefficients among adjacent blocks is calculated and the 64 bits authentication code is embedded by shifting this differential histogram during JPEG compression. The embedded code is extracted by reversing the shifting process and applied to check the integrity. Without any quality degradation, the original JPEG image can be reconstructed by calculating original quantized coefficients. The proposed algorithm was tested under various compression rates. The PSNR and compression ratio of the watermarked JPEG images were 32.33dB and 89.69% whose differences with the original JPEG images were 3.24dB and 2.58%.

Keywords: Reversible Data Hiding, Differential Histogram, Image Authentication, JPEG Compression.

Introduction
Multimedia contents can be easily generated and distributed with high computing performance hardware and software. As a result, the market of multimedia contents is increasing steadily. Intrinsically, digital contents can be modified and distributed without quality degradation. Therefore, they are vulnerable to digital forgery and illegal distribution and have raised copyright issues. In these days, multimedia security techniques are studied with the increase of its importance. Data hiding or digital watermarking can be an efficient solution to protect multimedia security and digital right management, which embeds confidential information called as the watermark into multimedia contents themselves. By retrieving the embedded watermark from the contents, it can be used for various purposes including ownership verification, copyright protection, and contents authentication, etc. Most data hiding methods only have focused on the robustness, which do not facilitate the recovery of the original content after the embedded watermark has been extracted [1]. In quality-sensitive applications such as medical, military, and artistic imaging, reversible data hiding is applicable, which can reconstruct the original image from the watermarked image after watermark removal [2-4]. However, most reversible data hiding methods have been studied for raw contents without any compression which is inevitable for multimedia contents. Therefore, they cannot be applicable for compressed contents.

In this paper, we propose a reversible data hiding algorithm that can be applicable to authenticate JPEG compressed images. During a JPEG compression procedure, the differential histogram of the quantized coefficients among adjacent blocks is calculated and the watermark is embedded by shifting this differential histogram. The embedded watermark is extracted by reversing the shifting process and used to authenticate. Also, the original JPEG image is reconstructed by calculating the original quantized coefficients. Differently from previous methods, the presented algorithm is applicable for JPEG compression which is commonly used for images and hence practically adapted to various applications. Also, the quality of the watermarked images does not degrade severely.

The paper is composed of as follows. In Sec. 2, related works are summarized. We present a novel JPEG authentication algorithm using reversible data hiding in Sec. 3. Experimental results are shown in Sec. 4 and Sec. 5 concludes.

Related Works
Reversible Data Hiding
Reversible data hiding is the method that can restore the original contents without any quality degradation after the embedded watermark is extracted. Many reversible data hiding researchers have studied to consider various characteristics of a given image to guarantee the perceptual transparency and reversibility. Celik et al. used a CALIC lossless compression algorithm and achieved a high capacity by using the generalized least significant bit embedding technique [5]. However, their performance differs depending on the structural characteristics of the images and varies according to the performance of the compression algorithm. Lee et al. applied an integer-to-integer wavelet transform to image blocks and embedded the message bits into the high-frequency wavelet coefficients of each block [6]. However, this method has high computational complexity as it is performed in the transform domain. Tian [5] and Alattar [6] proposed a difference expansion-based approach.
in which the differences between pixel pairs are considered to embed watermarks by expanding the differences. However, these methods have the burden of inserting a location map of all the possible expandable difference values with the watermark [5, 6]. Most of all, all these methods cannot be applied for JPEG compressed images, which is commonly used in multimedia applications.

To achieve high embedding capacity, Lin et al. inserted the watermark by modifying the histogram for each differential image after dividing the images into blocks [9]. Kim et al. inserted the watermark by shifting each differential histogram of the sub-sampled images that are acquired by calculating the difference between the reference sub-sampled image and the remaining sub-sampled images [3] Yeo et al. introduced a reversible data hiding method to embed the watermark by exploiting the gradual differential histogram of image blocks [10]. This method achieved the perfect reversibility by using the back-shifting scheme to the boundary pixel which could occur the overflow or the underflow of pixel values during the histogram shifting. Also, the JBIG compression scheme was applied for the back-shifting map to alleviate the store overhead. However, this method could not be applied to the JPEG image directly.

Choi et al. embedded the watermark into the data of the lossless compression step which is the second step of the JPEG compression procedure [11]. To minimize the visual quality degradation, they limited the watermark embeddable area of quantization coefficients in the watermark embedding process. However, since this method embedded the watermark into the compressed data of a single channel, it is difficult to apply to the JPEG compressed image directly, which has the chrominance data, the color information.

**JPEG Compression**

As shown in Figure 1, the JPEG compression procedure is composed of two parts: a lossy compression step and a lossless compression step. The lossy compression step consists of color-domain transform, DCT transform, and quantization. Most data loss during the JPEG compression happens in this step. The lossless compression step consists of run-length encoding and Huffman encoding that raise no data loss. Therefore, our JPEG authentication algorithm considers the data in the lossless compression step for data hiding.

**Proposed JPEG Authentication Using Reversible Data Hiding**

In general, the embedding capacity of reversible data hiding exploiting histogram shifting is determined to the largest value of the given histogram. In digital images, the statistical characteristics of adjacent areas are similar to each other. This similarity is called as the local similarity. We exploit this local similarity in the watermark embedding process.

The proposed reversible data hiding algorithm for JPEG compressed images exploits the differences of quantized DCT (QDCT) coefficients among the adjacent blocks. Our algorithm is composed of two processes: (1) watermark embedding and (2) watermark extraction and image reconstruction. The watermark embedding process is performed in the JPEG encoding step. On the other hand, the watermark extraction and image reconstruction process is performed in the JPEG decoding step. The explanation of calculating the QDCT coefficient differences is given in Sec. 3.1. The watermark embedding and extraction process is explained in Sec. 3.2 and Sec. 3.3, respectively.

**Calculating the QDCT Coefficient Differences**

In our JPEG authentication algorithm, we define one 16x16 block of a given image as one authentication unit. First, we generate the differential blocks $D_{b_n}$ by using four 8x8 QDCT blocks that composes one 16x16 authentication unit of a given image to exploit the local similarity. When these 4 QDCT blocks are defined as $D_0$, $D_1$, $D_2$, and $D_3$, respectively, the differences with $D_0$ and $D_1$, $D_2$, and $D_3$ are defined as $D_{b_1}$, $D_{b_2}$, $D_{b_3}$, respectively as shown in Equation (1).

$$D_{b_n}(i, j) = D_n(i, j) - D_0(i, j)$$ (1)

where $D_{b_n}$ is the difference between $D_n$ and $D_0$ and $n = 1, 2, 3$.

**Watermark Embedding**

During the lossy compression step, the quantization table is used to quantize the DCT coefficients. When the QDCT coefficients are modified by using this quantization table, the visual quality degradation of the JPEG compressed image is differently affected according to the location of the modified QDCT coefficients. We analyzed the amount of visual quality degradation when each QDCT coefficient is modified and this analyzed results are used to decide the embedding location. The analysis of the relation between the modified index of QDCT coefficients and the visual quality of several test JPEG compressed images is illustrated in Figure 2. We added 1 or 1 to 63 AC indices of 8x8 QDCT coefficients randomly to find out the relation between the modified index and the PSNR degradation. After the experiment, we found that most test images showed the similar tendencies of visual quality degradation when we modified the same index of QDCT coefficients. Therefore, the watermark embedding order in the QDCT coefficients is determined by the order of indexes which are less affected to the visual quality instead of using the zigzag order because the proposed embedding order has less visual quality degradation caused by embedding the watermark.

To embed the watermark signal into the QDCT coefficients, we first order the QDCT coefficients and the visual quality of several test JPEG images showed the similar tendencies of visual quality degradation. After the experiment, we found that most test images showed the similar tendencies of visual quality degradation when we modified the same index of QDCT coefficients. Therefore, the watermark embedding order in the QDCT coefficients is determined by the order of indexes which are less affected to the visual quality instead of using the zigzag order because the proposed embedding order has less visual quality degradation caused by embedding the watermark.

The index order of chosen AC components is as follow: 1, 2, 4, 5, 3, 7, 8, 9, 6, 12, 11, 13, 18, 17, 24, 22, 26, 30, 29, 33, 34, 38, 37, 36, 40, 41, and 42. Here, the index number is the same
as the index number of zigzag order. Thus, index 0 means the DC component and index 1 indicates the 1st AC component. The watermark is embedded into 3 differential blocks $D_{b_1}$, $D_{b_2}$, $D_{b_3}$ and the 1st QDCT block $D_0$. The watermark is embedded into differential blocks by the histogram shifting scheme which we have developed [2].

Watermark Extraction and Image Reconstruction
The watermark was embedded into the differential histogram of the QDCT coefficients by shifting the histogram. We first calculated 9 differential blocks by subtracting base QDCT block $D_0$ from their adjacent QDCT blocks in 3 channels. Then, the watermark was embedded into 9 differential blocks by using the histogram shifting.

$$Z_{DCT}(i) = \begin{cases} 
Z_{DCT}(i+1) & \text{if } Z_{DCT}(i) > 0 \\
Z_{DCT}(i) + W_m(j) & \text{if } Z_{DCT}(i) = 0 \\
Z_{DCT}(i) & \text{if } i = 0 \text{ or } Z_{DCT}(i) < 0
\end{cases}$$

where, $i$ is the index of zigzag order and $0 \leq j \leq \text{sizeof}(W_m) - 1$.

Since not only the watermark embedding and extraction processes but also the reconstruction of the QDCT blocks need un-watermarked $D_0$ data, the embedding order is very important. The embedding order of blocks is depicted in Figure 4. First, we embed the watermark into $D_{b_n}$ to generate the watermarked $D_{b_n}$, referring to $wD_{b_n}$. Then, we generate the watermarked $D_{b_n}$, referring to $wD_{b_n}$, by using the data of $wD_{b_n}$ and $D_{b_n}$. Last, we embed the watermark into $D_0$ to make the watermarked JPEG image.

Watermark Extraction and Image Reconstruction
The watermark was embedded into the differential histogram of the QDCT coefficients by shifting the histogram. We first calculated 9 differential blocks by subtracting base QDCT block $D_0$ from their adjacent QDCT blocks in 3 channels. Then, the watermark was embedded into 9 differential blocks by using the histogram shifting.

![Figure 2: Relation between the modified index of the QDCT block data and the PSNR in a test image.](image)

![Figure 3: Visiting order of differential blocks.](image)

![Figure 4: Watermark embedding order of target blocks.](image)

However, since the differential block $D_{b_n}$ is generated by using the un-watermarked base QDCT block $D_0$, we first extract the watermark from $wD_0$ and reconstruct $D_0$ from $wD_0$. Next, $wD_{b_n}$ are generated by using the reconstructed $D_0$ and $wD_{b_n}$. Then, we extract the embedded watermark from $wD_{b_n}$ and reconstruct $D_{b_n}$. Finally, we can reconstruct $D_1$, $D_2$, and $D_3$ by using these reconstructed $D_{b_n}$ and $D_0$. The overall reconstruction procedure is illustrated in Figure 5.
The way to extract the embedded watermark and to restore the original JPEG image can be modeled as Equation (3) and (4).

$$W_m(j) = \begin{cases} \text{skip} & \text{if } i = 0 \\ \text{ZDCT}_w(i) & \text{if } \text{ZDCT}_w(i) \in \{0, 1\} \\ \end{cases}$$  \tag{3}

$$\text{ZDCT}_w(i) = \begin{cases} \text{ZDCT}_w(i) & \text{if } i = 0 \text{ or } \text{ZDCT}_w(i) < 0 \\ \text{ZDCT}_w(i) - 1 & \text{if } \text{ZDCT}_w(i) > 0 \\ \end{cases}$$  \tag{4}

where, \(i\) is the index of zigzag order and \(0 \leq j \leq \text{sizeof}(W_m) - 1\). The extracted watermark is used to authenticate the JPEG images.

**Experimental Results**

We evaluate the performance of the proposed algorithm by using 24 bits images of 512×512 pixels from USC-SIPI (University of Southern California-Signal & Image Processing Institute) as shown in Figure 6. The 64 bits watermark is embedded into each block of 16×16 pixels.

![Figure 6: Test images from USC-SIPC database.](image)

First, we evaluate the perceptual invisibility of the proposed algorithm. To test invisibility, we compared the perceptual quality between the standard JPEG images and watermarked JPEG images using the proposed algorithm. The perceptual quality was measured with peak signal-to-noise ratio (PSNR).

$$\text{PSNR}_{dB} = 10 \cdot \log_{10} \frac{255^2}{\text{MSE}}$$

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (O(i, j) - W(i, j))^2$$  \tag{5}

where \(O(i, j)\) is the intensity of the original image and \(W(i, j)\) is the intensity of the watermarked image.

Test results are shown in Table 1 and Figure 7. As shown in the results, the visual quality of watermarked JPEG images was slightly lower than standard one. However, the difference was not so high. Moreover, the un-watermarked JPEG images could be reconstructed after watermark extraction because the proposed reversible algorithm can remove the embedded watermark and recover the original JPEG compression coefficients at 100%.

<table>
<thead>
<tr>
<th>Image</th>
<th>Standard JPEG compression</th>
<th>Proposed JPEG compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>37.68 dB</td>
<td>33.29 dB</td>
</tr>
<tr>
<td>Baboon</td>
<td>31.64 dB</td>
<td>30.13 dB</td>
</tr>
<tr>
<td>House</td>
<td>35.66 dB</td>
<td>32.19 dB</td>
</tr>
<tr>
<td>Lena</td>
<td>36.34 dB</td>
<td>32.75 dB</td>
</tr>
<tr>
<td>Pepper</td>
<td>35.16 dB</td>
<td>32.19 dB</td>
</tr>
<tr>
<td>Sailboat</td>
<td>33.55 dB</td>
<td>31.24 dB</td>
</tr>
<tr>
<td>Splash</td>
<td>38.42 dB</td>
<td>33.95 dB</td>
</tr>
<tr>
<td>Tiffany</td>
<td>36.13 dB</td>
<td>32.88 dB</td>
</tr>
<tr>
<td>Average</td>
<td>35.57 dB</td>
<td>32.33 dB</td>
</tr>
</tbody>
</table>

![Figure 7: Perceptual quality comparison between standard JPEG compression and our watermarked JPEG compression.](image)

Since the proposed algorithm is designed for JPEG images, the comparison of compression ratios before and after applying the proposed reversible data hiding algorithm could be one criterion of the performance measure. Compression ratios shown in Table 2 were calculated by using the ratio of the size of an original image formatted in BMP to the size of a JPEG compressed image including the header information. As shown in the results, the differences between the compression ratios of the standard JPEG compressed images and those of the watermarked JPEG images were very small. Thus, we can conclude that the proposed algorithm performs well in the aspect of compression ratios.
In the watermark embedding process, we generated image authentication algorithm using reversible data hiding. The contribution of the paper is designing a reversible data hiding algorithm that can be applicable for JPEG compression, the additional study for JPEG2000 compression of a standard JPEG compression method, it is utilized as an add-on of standard JPEG compression. Since the presented algorithm was focused on JPEG images. Since the proposed algorithm is designed in the base domain, the quality of the watermark JPEG images should be studied. Also, the enhancement of the quality of the watermark JPEG images should be studied. Since the presented algorithm was focused on JPEG compression, the additional study for JPEG2000 compression will be required.

### Conclusion

Multimedia contents can be illegally manipulated without quality degradation and that raises the needs to ensure the integrity of the contents. This paper presented a novel JPEG image watermarking based on integer-to-integer wavelet transform, IEEE Transactions on Circuits and Systems for Video Technology, 13(8), pp. 890-896.

#### Table 2: Compression ratio comparison (unit : bytes)

<table>
<thead>
<tr>
<th>Image</th>
<th>BMP</th>
<th>Standard compression</th>
<th>JPEG compression</th>
<th>Proposed compression</th>
<th>JPEG compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>786,488</td>
<td>48,961</td>
<td>93.77%</td>
<td>69,564</td>
<td>91.16%</td>
</tr>
<tr>
<td>Baboon</td>
<td>786,488</td>
<td>104,711</td>
<td>86.69%</td>
<td>127,001</td>
<td>83.85%</td>
</tr>
<tr>
<td>House</td>
<td>786,488</td>
<td>64,215</td>
<td>91.84%</td>
<td>85,461</td>
<td>89.13%</td>
</tr>
<tr>
<td>Lena</td>
<td>786,488</td>
<td>49,008</td>
<td>93.77%</td>
<td>69,296</td>
<td>91.19%</td>
</tr>
<tr>
<td>Pepper</td>
<td>786,488</td>
<td>55,831</td>
<td>92.90%</td>
<td>76,796</td>
<td>90.24%</td>
</tr>
<tr>
<td>Sailboat</td>
<td>786,488</td>
<td>70,265</td>
<td>91.07%</td>
<td>91,984</td>
<td>88.30%</td>
</tr>
<tr>
<td>Splash</td>
<td>786,488</td>
<td>42,574</td>
<td>94.59%</td>
<td>89,807</td>
<td>92.40%</td>
</tr>
<tr>
<td>Tiffany</td>
<td>786,488</td>
<td>50,719</td>
<td>93.55%</td>
<td>68,876</td>
<td>91.24%</td>
</tr>
<tr>
<td>Average</td>
<td>786,488</td>
<td>60,786</td>
<td>92.27%</td>
<td>81,098</td>
<td>89.69%</td>
</tr>
</tbody>
</table>

### Acknowledgement

This paper was supported by Research Fund, Kumoh National Institute of Technology.

### References