Quantization-Based Image Steganography without Data Hiding Position Memorization

Yusuke SEKI*, Hiroyuki KOBAYASHI†, Masaaki FUJIYOSHI* and Hitoshi KIYA*
*Department of Electrical Engineering, Tokyo Metropolitan University, Hachioji-shi, Tokyo 192–0397, Japan
{yusuke@isys., fujiyoshi@, kiya@}eei.metro-u.ac.jp
†Department of Electrical Engineering, Tokyo Metropolitan College of Technology, Shinagawa-ku, Tokyo 140–0011, Japan
hkob@tokyo-tmct.ac.jp

***Abstract*— This paper proposes a quantization-based steganography method of extracting hidden data without any reference images or memorization of positions, into which data are embedded. Since the proposed method offers the user the flexibility of choosing data hiding positions, it enables the user to select positions for embedding data on an individual image basis and/or the basis of the coding scheme being applied to the images. Simulation results show the effectiveness of this method.**

I. Introduction

Digital images flood our lives through various media with the growing popularity of information terminals. Most images are compressed in compliance with coding standards for transmission and storage: JPEG [1] and JPEG 2000 [2] are used for still images compression, and MPEG [3], [4], [5] is widely used to encode video sequences. Simultaneous processing of images and their associated data is often desired for value added functionalities. Adding text data such as a caption to the image makes it possible to search desired images from a huge image database [6], [7], and monitoring and/or estimation of the image-quality at the receiver side [8] as well as detection and/or correction of channel errors [9] are achieved by embedding adequate data into images.

Several methods have been proposed to compound a compressed image with associated data in a transformed domain. A method is referred to as *oblivious* if no reference images are needed for extraction. This paper proposes an oblivious steganography. Oblivious data hiding methods dedicated to the compressed images enable the embedding of integer data into integer coefficients. Whereas, the other oblivious methods are not for the compressed images. This paper proposes a method classified as the former type, which has the advantage that hidden data are not distorted by the compression process. Furthermore, hidden data are extracted directly from the transformed domain, i.e., no decoding process is required for data extraction.

Conventional oblivious methods dedicated to compressed images, however, still require to memorize positions for data hiding to extract the hidden data. Moreover, since they fix the positions within the whole image, it is difficult to choose data hiding positions on an individual image basis and/or the basis of the coding scheme being applied to the images. On the other hand, methods that embed only one bit per one position are not able to embed multiple bits in any one position [10]. Methods that are able to be extended to embed multiple bits in one position may degrade the image-quality considerably, because they have to embed L-level data using $\lfloor \log_2 L \rfloor$ bit data, e.g., four bits are hidden for data that has nine levels [6], [7].

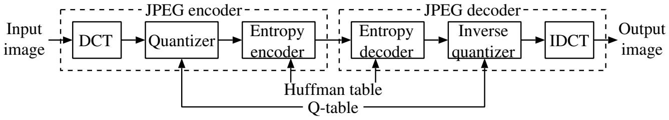

Fig. 1. JPEG codec [1].

This paper proposes an oblivious image steganography method that also requires no knowledge on data hiding positions when the data are extracted. The proposed method offers the user the flexibility of choosing positions to hide data. It, thus, modifies the coefficients to embed data on the basis of the individual image and/or image compression technology. Since it hides integer data in transformed coefficients after the quantization process of image compression, the hidden data are no longer distorted by the rest of the compression process. The proposed method hides $\log_2 L$ bits rather than $\lfloor \log_2 L \rfloor$ bits for L-level data into one particular area. It, thus, reduces image-degradation more than with conventional methods.

II. Steganography for Coded Images

Conventional steganography for JPEG coded images and problems associated with it are discussed in this section to clarify the aim of the proposed steganography.

A. JPEG and Integer Coefficients

A block diagram of JPEG coder is shown in Fig. 1 [1]. An image is firstly divided into an 8×8 pixels-sized block and a discrete cosine transformation (DCT) is applied to each block. All coefficients in a block are quantized as integers using a *Q-table* [1]. Quantized coefficients sorted by the *zigzag scan* order [1], then, are encoded using a Huffman encoder. Zeros lasting to the last coefficient ordered by the zigzag scan are replaced by an End of Block (EOB) marker, rather than being encoded, to improve the coding efficiency.

In the rest of this paper, the k-th DCT coefficient in the zigzag scan order in one 8×8 coefficients-sized block is represented by a_k, where $k = 0, 1, \ldots, 63$. Dividing a_k by q_k, a_k's corresponding quantization step defined in a Q-table,

0-7803-8834-8/05/$20.00 ©2005 IEEE.

introduces quotient b_k, and the integer rounded off for b_k is quantized coefficient c_k.

For data to be hidden in integer coefficient c_ks that are input to a Huffman encoder, the data need to be an integer.

B. Conventional JPEG steganography

Several conventional steganography methods and their main features are briefly mentioned in the following sections. All methods embed data in a transformed domain.

1) Replacing the Least Significant Bit with One-Bit Data [10]: This method embeds one bit through the quantization of a_k where $k = p$. Quantized coefficient c_k corresponding to a_k is set to the closest even integer to hide zero and is set to the closet odd integer to hide one [10]. The least significant bit (LSB) of c_k, thus, is substituted for one-bit data. There are three disadvantages to this method. The first is that parameter p is required to extract hidden data. The second is the fixed p used in all blocks. The third is that the maximum amount of bits per hiding place is one.

2) Replacing the Highest Frequency Coefficient with Hidden Data and Modifying Q-table [6]: The highest frequency quantized coefficient c_{63} itself, instead of the LSB of c_{63}, is substituted for hidden data in this method [6]. To improve the image-quality of decoded stego images, q_{63} is substituted by one in the Q-table after embedment and c_{63} is replaced by zero before decoding the stego images. This method enables the hiding of multiple bits in one position. The first problem with this method is c_{63} is used for data hiding in all blocks. The second is that the size of the stego JPEG codestream increases greatly. The third is the requirement that a proprietary JPEG decoder replaces c_{63} with zero.

3) Replacing Least Significant Bits with Hidden Data [7]: This method embeds n-bit data into c_k, where $k = p$, by replacing n of LSB's with n-bit data [7]. This method enables the hiding of multiple bits in one position. There are two disadvantages to this method. First, fixed p is used in every blocks, and the second, p is required for data extraction.

III. The Proposed Method

In this section, an embedment and extraction algorithms are described using a JPEG coded image as an example. It is assumed that data sequence $\mathbf{w}$ is hidden in a JPEG coded image that consists of M of 8×8 pixels-sized blocks. Sequence $\mathbf{w}$ consists of M of elements and is represented as $\mathbf{w} = \left\{ w_m \mid m = 1, \ldots, M \right\}$. Data element w_m is hidden in the m-th block and each w_m has L levels from zero to $L-1$. This method hides $\log_2 L$ bits rather than $\lfloor \log_2 L \rfloor$ bits for w_m.

A. Data Hiding in a Compressed Domain

The hiding algorithm in the proposed method is described as hiding w_m in an m-th block after quantization of the transformed coefficients. In practice, this process is repeated M times to hide whole $\mathbf{w}$.

This algorithm firstly modulates data w_m to d, and this paragraph describes how it is modulated. Quantized coefficients in the m-th block are summed up by

$$S = \sum_{k=0}^{63} c_k, \tag{1}$$

where c_k represents the quantized coefficients. Positive remainder r, then, is obtained by dividing S by L as

$$r = S \bmod L, \quad r > 0. \tag{2}$$

The difference d_1 between the obtained r and data to be hidden w_m is calculated directly by

$$d_1 = |w_m - r|, \tag{3}$$

and another difference d_2 is also given as

$$d_2 = L - d_1, \tag{4}$$

because of the modulus rule. Finally, the modulated value d that is actually hidden into the m-th block is given by

$$d = \min(d_1, d_2). \tag{5}$$

Actual data hiding is achieved by modifying one or several coefficients in the m-th block so that the summation of coefficients that is represented by $\hat{S}$ satisfies

$$\hat{S} = \begin{cases} S-d, & r > w_m \text{ and } d_1 < d_2 \\ S+d, & r > w_m \text{ and } d_1 \geq d_2 \\ S+d, & r < w_m \text{ and } d_1 < d_2 \\ S-d, & r < w_m \text{ and } d_1 \geq d_2 \\ S, & r = w_m \end{cases}. \tag{6}$$

It is noted again that this algorithm allows us to modify either one or several coefficients to hide data. That is, the proposed steganography allows the user the flexibility of choosing the coefficient for data hiding. The k-th stego coefficient is represented by $\hat{c}_k$ hereafter, whether $c_k = \hat{c}_k$ or not.

B. Data Extraction in a Compressed Domain

The extraction of w_m from the m-th block is described in this section, and it is repeated M times to extract the whole of $\mathbf{w}$ in practice. An entropy decoding process is applied to a stego JPEG codestream and, then, hidden data are extracted without using the inverse quantization process.

The stego coefficients $\hat{c}_k$s are summed up to obtain $\hat{S}$ by

$$\hat{S} = \sum_{k=0}^{63} \hat{c}_k. \tag{7}$$

The remainder $\hat{r}$, then, is obtained by dividing $\hat{S}$ by L as

$$\hat{r} = \hat{S} \bmod L, \quad \hat{r} > 0. \tag{8}$$

This remainder $\hat{r}$ is identical to w_m, so w_m can be extracted without decoding the whole of a stego JPEG codestream.

It is noteworthy that this extraction algorithm requires only one parameter L. No knowledge of positions for data hiding is required in this proposed steganography, whereas conventional methods require such information. Moreover, lossless processing is applied after quantization of coefficients in JPEG.

Hidden data, thus, is extracted error free. Since modifying several c_ks to hide data does not affect the structure of the JPEG codestream, a stego JPEG codestream is decodable with a standard JPEG decoder.

C. Features of the Proposed Method

The most important feature of the proposed method is that fixed coefficients for data hiding do not have to be set. This feature gives the proposed method the following advantages:

- No knowledge of coefficients for hidden data is required in any data extraction process
- The coefficients are chosen depending on the individual image and/or coding technique.

The latter advantage allows us to choose coefficients taking into consideration the reduction of degradation in coding efficiency, the compensation of quantization errors, and the improvement in the subjective imperceptibility of hidden data.

In addition, the proposed method enables the hiding of data in images encoded in accordance with any of the compression standards. Furthermore, since this proposed method embeds $\log_2 L$ bits rather than $\lfloor \log_2 L \rfloor$ bits in embedding data in which each element has L levels, it reduces the image-quality deterioration better than conventional steganography methods.

D. Examples of Coefficient Modification

As described above, the proposed method offers the user the flexibility of choosing the coefficients to be modified to hide data. Two example strategies for modifying coefficients to hide data are described in this section. The former is the compensation of quantization errors in the JPEG encoding process, and the latter is the suppression of an increase in size of the JPEG codestream.

1) Quantization Error Compensation: Quantization errors can occur in each of the coefficients in a block in the JPEG encoding process. The error in the k-th coefficient is given by

$$e_k = b_k - c_k, \tag{9}$$

$$c_k = \text{round}\,(b_k)\,, \tag{10}$$

where e_k and b_k represent a quantization error satisfying $-0.5 < e_k \leq 0.5$ and the quotient of dividing the original coefficient a_k by the corresponding quantization step q_k, respectively. If the magnitude of e_k is reduced, the image-quality of the decoded image improves. In this strategy, thus, data are hidden so that the magnitude of e_k is reduced.

Before c_ks are modified, quantization errors e_ks are obtained and indicated ks are sorted as e_ks in descending order. Sorted ks are represented as $k(i)$ where $i = 1, \ldots, 63$. $k(0)$ and $k(63)$ indicate the frequency corresponding to the maximum e_k and the minimum e_k, respectively. Then, under the conditions that $\hat{S} = S + d$ in Eq. (6), the following process is carried out:

```
1: $i \leftarrow 0$
2: while $i < d$ do
3:     $c_{k(i)} \leftarrow c_{k(i)} + 1$
4:     $i \leftarrow i + 1$
5: end while
```

However, when $\hat{S} = S - d$ in Eq. (6), step 3 in the algorithm described above is replaced by

```
3:     $c_{k(63-i)} \leftarrow c_{k(63-i)} - 1$
```

This strategy hides data by adding ones to the d of c_ks that correspond to the largest e_ks or by subtracting ones from the d of c_ks that correspond to the smallest e_ks so that the algorithm maximizes compensation in distortion caused by quantization errors. Consequently, the image-quality of a stego JPEG encoded image improves. It is noted that this algorithm chooses $c_{k(0)}$ or $c_{k(63)}$, whichever are greater in terms of magnitude, for hiding one-bit data.

2) Minimizing the Increase in the Size of Codestream: As mentioned in Sect. II-A, the greater the number of zeros lasting to the last coefficient, the more the coding efficiency improves. The last non-zero coefficient, thus, is chosen for hiding data to minimize the size of the stego codestream in this strategy.

For this strategy, if Eq. (6) requires $S + d$, the following algorithm is used.

```
1: $k \leftarrow 63, j \leftarrow 0$
2: while $k \geq 0$ do
3:     while $(j < d)$ and $(c_k < 0)$ do
4:         $c_k \leftarrow c_k + 1$
5:         $j \leftarrow j + 1$
6:     end while
7:     $k \leftarrow k - 1$
8: end while
```

If Eq. (6) is produced by $S - d$, Steps 3 and 4 are replaced by

```
3:     while $(j < d)$ and $(c_k > 0)$ do
4:         $c_k \leftarrow c_k - 1$
```

This strategy makes the last non-zero coefficient to be zero so that it is the coding efficiency rather than the image-quality that improves.

IV. Simulations

In this section, proposed method A described in Sect. III-D.1 and B described in Sect. III-D.2 are compared with conventional method A [6] described in Sect. II-B.2 and B [7] described in Sect. II-B.3, in terms of the image-quality of the decoded stego image and the size of the stego JPEG codestream. Conventional method A involves replacing c_{63} before decoding the stego codestream. Conventional method B involves choosing c_5 for data hiding because q_5 is the smallest in the standard Q-table, i.e., the best image-quality results.

The selected cover image is the well known grayscale "Lena" image whose size is 512×512. Data $\mathbf{w}$ consists of equiprobable ones and zeros and the length of $\mathbf{w}$ is 8192. That is, two-bit data is hidden into one 8×8 pixels-sized block. The image-quality controlling parameter, Q-factor, in the used JPEG codec (PVRG-JPEG) is set between 20 and 200.

Figure 2 shows the peak signal-to-noise ratio (PSNR) between the JPEG coded images and the input noncoded image. The coded images are the standard JPEG coded images conveying no data and the stego JPEG images conveying 8192-bits hidden data. Each stego image series is generated using proposed method A, proposed method B, conventional method A, or conventional method B. Whereas conventional method

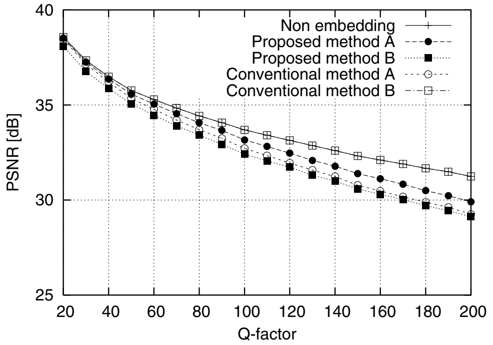

Fig. 2. The PSNR between JPEG images and the original image versus Q-factor (Hidden data: 2 bits/block).

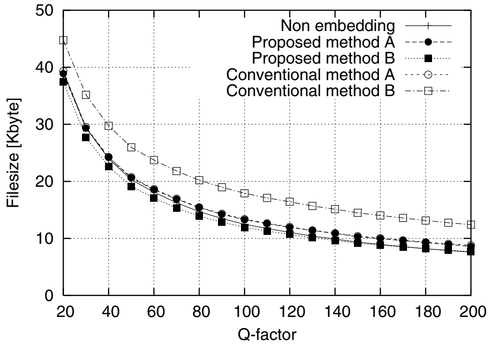

Fig. 3. Filesize of JPEG images versus Q-factor (Hidden data: 2 bits/block).

A generates stego images whose quality is quite similar to the standard JPEG coded images, it increases the size of the stego codestream greatly as shown in Fig. 3. Figure. 4 also shows the size of the stego codestreams, in terms of the stego codestream size-to-the standard codestream size ratio, instead of the actual size. Proposed method B enables a reduction in size in comparison to that of the standard codestream.

To evaluate performance taking into consideration both the image-quality and the codestream-size, another index is introduced. That is the PSNR-to-filesize ratio. It tells us the degree of one KiB data in the codestream's contribution to the image-quality, and is shown in Fig. 5. Though the proposed method B hides 8192-bits data into a stego codestream, it degrades performance slightly in comparison to the standard codestream conveying no data, shown in Fig. 5.

V. Conclusions

This paper has proposed a quantization-based steganography method that requires no knowledge of the positions of hidden data in the extraction process. Since the proposed method offers the user the flexibility of choosing positions to hide data, positions can be chosen in accordance with the individual image and/or coding technique. Simulations using JPEG as the coding technique show the effectiveness of the proposed method. This method can also be applied to any coding technique.

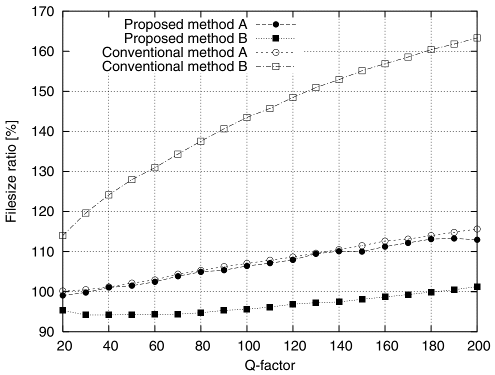

Fig. 4. The stego JPEG images-to-the original JPEG image filesize ratio versus Q-factor (Hidden data: 2 bits/block).

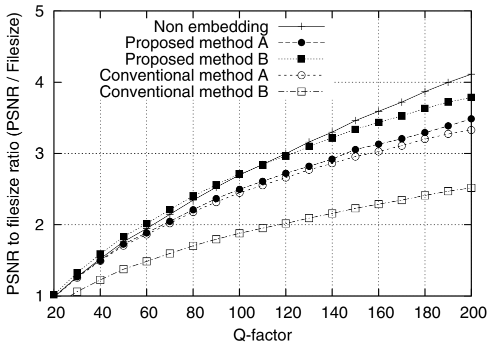

Fig. 5. The PSNR-to-filesize ratio versus Q-factor (Hidden data: 2 bits/block).

References

[1] *Information technology — Digital compression and coding of continuous-tone still image: Requirements and guidelines*. Int. Std. ISO/IEC IS-10918-1, 1994.

[2] *Information technology — JPEG 2000 image coding system – Part 1: Core coding system*. Int. Std. ISO/IEC IS-15444-1, Dec. 2000.

[3] *Information technology — Coding of moving pictures and associated audio for digital storage media up to about 1,5 Mbits/s — Part 2: Video*. Int. Std. ISO/IEC IS-11172-2, 1993.

[4] *Information technology — Generic coding of moving pictures and associated audio information: Video*. Int. Std. ISO/IEC IS-13818-2, 2000.

[5] *Information technology — Coding of audio-visual objects — Part 2: Visual*. Int. Std. ISO/IEC IS-14496-2, 2001.

[6] Y. Noguchi, H. Kobayashi, and H. Kiya, "A method of extracting embedded binary data from JPEG bitstreams using standard jpeg decoder," in *Proc. IEEE ICIP*, 2000.

[7] M. Fujiyoshi and H. Kiya, "A data hiding method for indexing JPEG-coded images and theoretical analyses of image-quality," in *Proc. ITC-CSCC*, 2004.

[8] L.D. Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, and G. Depovere, "Implementation of a real-time digital watermarking process for boradcast monitoring on a TriMedia VLIW processor," *IEE Proc. Visual and Image Signal Processing*, vol.147, pp.371–376, Aug. 2000.

[9] D.L. Robie and R.M. Mersereau, "Video error correction using steganography," *EURASIP J. Appl. Signal Processing*, vol.2002, pp.164–173, Feb. 2002.

[10] P.H.-W. Wong and O.C. Au, "Data hiding and watermarking in JPEG compressed domain by DC coefficient modification," in *Proc. SPIE*, vol.3971, pp.237–244, 2000.