WATERMARKING FOR JPEG IMAGE AUTHENTICATION SURVIVING INTEGER ROUNDING IN DECOMPRESSION

Hiroshi Ito, Koichi Magai, Ryousuke Fujii, and Mitsuyoshi Suzuki

Mitsubishi Electric Corporation
Information Technology R&D Center
5-1-1 Ofuna, Kamakura-city, Kanagawa, Japan 247-8501

ABSTRACT

A watermarking scheme for JPEG image authentication is proposed. It is a direct extension of Wong's algorithm to JPEG-coded images where the signature is embedded at the end of scanned DCT coefficients, instead of LSB's of raw pixel values. We address a problem that the watermark disappears after the integer rounding in JPEG decompression and show that imposing a restriction on the quantization step sizes for DCT coefficients can solve this problem. To avoid the limitation on compressed picture quality set by this restriction, we introduce an embedding technique to use different quantization vectors for the watermarking and the JPEG compression. Simulation results are shown to verify the proposed scheme.

1. INTRODUCTION

Due to simplicity of manipulating digital images, it has been recognized that authentication of images is important in some application areas[1]. Watermarking is a promising technology since all the information to verify the authenticity is coded together with the image and no handling of separate information is necessary.

Wong and Memon [2] have proposed a scheme to embed a hashing result of pixel values XORed with a checking pattern followed by the private key encryption. However, their scheme embedded the information in the LSB's of images and is therefore not applicable to JPEG compressed images. Some data structures of quantized DCT coefficients can be used to verify the authenticity of compressed images. Lin and Chang[3] proposed to record the magnitude relationships between spatially separate DCT coefficients as an image fingerprint. However, it seems possible to modify the image so that the magnitude relationships remain unchanged. Their scheme is considered to be among those techniques[1, 4] that embed robust or semi-fragile watermarks and detect any manipulation when the watermark is lost. In such schemes, changes to the image are detected only probabilistically.

We directly apply Wong's scheme to JPEG compressed images. The structure of quantized DCT coefficients are computed and then embedded at the end of each DCT block. With this approach, the authenticity is guaranteed precisely within the quantization step sizes. Then we discuss the problem of watermark loss in the JPEG decompression. The loss is caused partly by the integer rounding after the inverse DCT. We attack this problem by imposing a restriction on the quantization step sizes. Another possibility of watermark loss comes from confining decoded pixel values to a predetermined range typically of [0:255]. In this paper we only address the integer rounding problem while some results on the latter appears in [5]. Since the above restriction limits the quality of JPEG-coded images, we propose a scheme to use different quantization tables for the watermarking and the JPEG-compression.

The rest of the paper is organized as follows. Section 2 describes the watermarking algorithm. Section 3 discusses a scheme to make the watermark survive the JPEG decoding. Section 4 talks about how the scheme can be extended to allow high quality compression. Section 5 shows some computer simulation results. Section 6 gives concluding remarks.

2. ALGORITHM

Before discussing the watermarking algorithm, we clarify the identity of two images in the JPEG compressed domain. With the following definition, identical images need not match precisely in the pixel domain but are required to match in the quantized DCT domain.

Definition 1 *Two images are defined to be identical if the corresponding quantization indices of DCT coefficients coincide perfectly when the same quantization table is applied.*

0-7803-8554-3/04/$20.00 ©2004 IEEE.

A watermarking algorithm to verify the integrity based on the above definition is constructed as follows. A group of K DCT blocks is first generated in which K bits of signature are embedded on one bit per block basis. The value of K is typically 128. The signature is generated in the following steps.

1. For each block, compute run-length code of quantized DCT coefficients appending a dummy run after the end of the scanning. Then concatenate K of them to get a structure of DCT coefficients.

2. Compute a hash value of the structure, which is of length K bits.

3. As in Wong's algorithm[2], take bit-wise XOR of the hash with a binary bit pattern and then encrypt it with a private key.

Each bit of the signature is embedded by changing the quantization index of the last DCT coefficient in the run-length coding to -1 or 1 depending on the value of the bit. Here, it is important that the marked value takes a unique number of either -1 or 1 to guarantee the detection of any change to these coefficients. Note that there would be possibility to modify the image if we have used a normal quantization technique having multiple bins for the same bit value.

With the above watermark, it is easily shown that the authenticity can be verified by checking if the extracted signature matches exactly the one computed from the image data. If either a signature bit or a data bit is modified, the two values do no match. They match iif both part have not been modified.

3. SURVIVAL AFTER DECOMPRESSION

Watermark loss in JPEG decompression occurs partly in the rounding process of decoded pixel values. Since the inverse DCT produces real numbers, they must be converted to integers to be represented in a limited number of resolutions.

We will explain how the watermark disappears in the integer rounding referring to Fig. 1. It shows a vector space for the two dimensional DCT, where black circles represent integer vectors in the spatial domain while white circles are quantized vectors in the transform domain. The quantization boundaries are shown by solid lines. We call a region separated by the boundaries a bin. White circles are quantization representatives of these bins. Now if we look at the grayed bin i, it has two input vectors x_1^i and x_2^i both of which are transformed and quantized to y^i. However, once y^i is inverse-transformed and rounded to the nearest integer vector, it is decoded to x_1^j in bin j, not in bin i. Then,

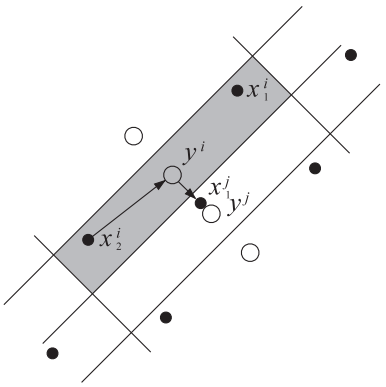

Fig. 1. A case where the watermark disappears

if x_1^j is transformed and quantized again, the vector comes to y^j which is different from y^i. Hence the DCT structure is changed and the watermark disappears after decompression.

However, we see that if the quantization is coarse enough, a large number of integer vectors around the representing vector would come to belong to the same bin and that the watermark loss would be hard to happen. Assuming that uniform quantization is used for all the DCT coefficients and that the decoded pixel values after the inverse DCT are rounded to the nearest integers, we have the following theorem.

Theorem 1 *If $q_i > \sqrt{N}$ is satisfied for all i's, the vectors of inverse-transformed DCT coefficients before and after integer rounding belong to the same bin, where q_i is the quantization step size for the i-th DCT coefficient and N is the dimension of the transform.*

Proof Let $c = \{c_1, c_2, \cdots, c_N\}$ be a real-valued decoded vector after inverse-DCT. Without loss of generality, we assume that $0 \le c_i \le 1$ $(i = 1, \cdots, N)$. Let $x = \{x_1, x_2, \cdots, x_N\}$ be an integer vector for which $x_i = 0$ or $x_i = 1$. We have 2^N such vectors. Let $\hat{x} = \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_N\}$ be the nearest integer vector to c. Then we have for all i's,

$$(c_i - \hat{x}_i)^2 \le (1/2)^2 \tag{1}$$

and therefore

$$|c - \hat{x}| = \sqrt{\sum_{i=1}^{N} (c_i - \hat{x}_i)^2} \le (1/2)\sqrt{N}, \tag{2}$$

where $|\cdot|$ represents the vector norm. The above inequality and the condition that $q_i > \sqrt{N}$ yields

$$|c - \hat{x}| \le (1/2)\sqrt{N} < (1/2)q \tag{3}$$

where q is the minimum of q_i over $0 \le i \le N$. On the other hand, a vector x which satisfies

$$|c - x| \le (1/2)q \tag{4}$$

is quantized to c since, if the above inequality holds, we also have

$$|c_i - x_i| \le (1/2)q \le (1/2)q_i. \tag{5}$$

This proves that vector $\hat{x}$ belongs to the same bin as c.

4. WATERMARKING FOR HIGH QUALITY COMPRESSION

The restriction on quantization step sizes in Theorem 1 may limit the picture quality of JPEG-coded images. In this section we will introduce a method to use separate quantization tables for compression and watermarking, by which high quality JPEG compression is made possible.

Let $q^c = \{q_1^c, \cdots, q_N^c\}$ and $q^w = \{q_1^w, \cdots, q_N^w\}$ be the quantization vectors for compression and watermarking respectively. Let $N = 64$. Then, for $i = 1, \cdots, 64$, the following conditions must be met

$$q_i^w > \sqrt{64} = 8, \tag{6}$$
$$q_i^c \le q_i^w. \tag{7}$$

The former condition comes from Theorem 1 while the latter is necessary to make the embedding always possible. To meet these conditions, we simply choose q_i^c arbitrarily and get q_i^w by

$$q_i^w = \max(8, q_i^c). \tag{8}$$

Watermarks are embedded as described in Section 2 using q^w and we get $\hat{y}^w$ as an index vector of quantized DCT coefficients. $\hat{y}^w$ is a perfectly compliant JPEG expression and is actually encoded in the bit stream if $q^w = q^c$. However, for high fidelity compression, $\hat{y}^c$ is encoded instead as a vector which minimizes

$$d = |Q_c^{-1}(\hat{y}^c) - T(x)|, \tag{9}$$

subject to

$$Q_w(Q_c^{-1}(\hat{y}^c)) = \hat{y}^w, \tag{10}$$

where x is the input vector to be coded and $T(\cdot)$, $Q(\cdot)$, and $Q^{-1}(\cdot)$ are transform, quantization and inverse-quantization functions respectively. The vector $\hat{y}^c$ can be simply computed as

$$\hat{y}_i^c = \min\{\hat{y}_{i\max}^c, \max\{\hat{y}_{i\min}^c, Q_c(y_i)\}\}, \tag{11}$$

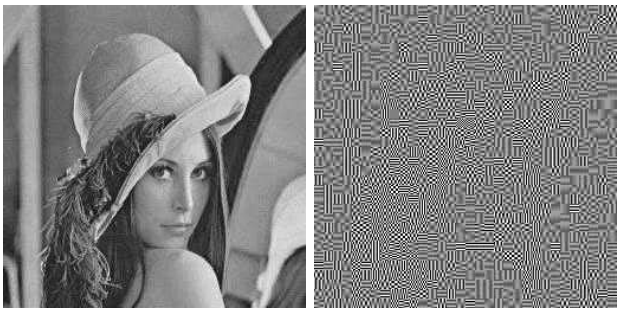

Fig. 2. Watermarked image (left) and embedded pattern (right)

where $\hat{y}_{i\min}^c$ and $\hat{y}_{i\max}^c$ are the minimum and maximum indices of $\hat{y}_i^c$ satisfying (10) with y_i the i-th DCT coefficient.

Although the watermark information is preserved in $\hat{y}^c$, normal JPEG decoding can lose this information since $\hat{y}^c$ may not be at the center of a quantization bin of q^w not equal to q^c. However we can always find an integer vector which keeps the watermark since we can compute $\hat{y}^w$ using (10) and it is guaranteed from Theorem 1 that there must be at least one integer vector in the same bin. After setting

$$x^t = T^{-1}(Q_w^{-1}(\hat{y}^w)) \tag{12}$$
$$x^0 = \mathrm{int}(T^{-1}(Q_c^{-1}(\hat{y}^c))), \tag{13}$$

with $\mathrm{int}(\cdot)$ the integer rounding function, the following algorithm finds such a vector.

Algorithm 1

Step 1 $i \leftarrow 0$.

Step 2 If $Q_w(T(x^i)) = \hat{y}^w$ then stop.

Step 3 $j \leftarrow \arg\max_j |x_j^t - x_j^i|$

Step 4 For $k = 1, \cdots, N$;

$$x_k^{i+1} \leftarrow \begin{cases} x_k^i + \frac{x_j^t - x_j^i}{|x_j^t - x_j^i|} & k = j \\ x_k^i & k \ne j \end{cases}$$

Step 5 $i \leftarrow i + 1$. Go to Step 2.

In the above algorithm, x^0 is modified repeatedly toward x^t until it holds embedded information. The algorithm always converges since, in each repetition, $|c_t - x_{i+1}| < |c_t - x_i|$ is always satisfied.

5. EXPERIMENTAL RESULTS

Fig. 2 shows a watermarked LENA image and the embedded watermark signal with contrast enhanced. The

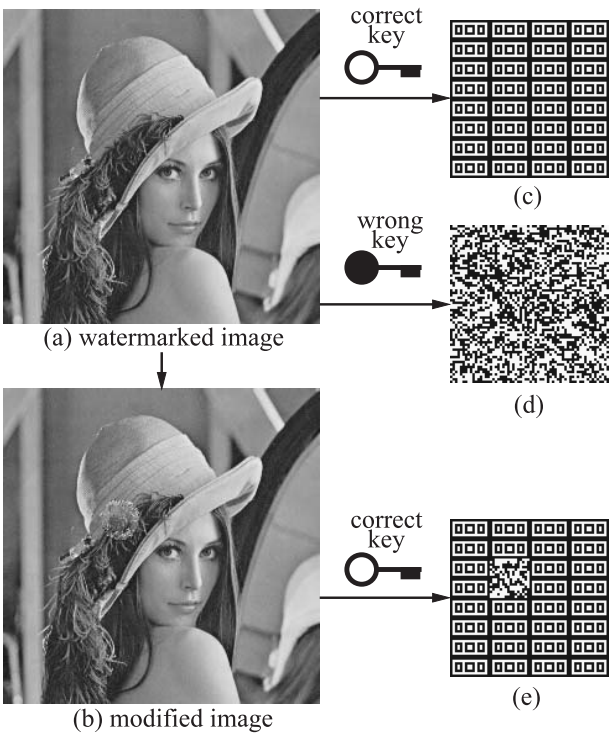

Fig. 3. Verification of authenticity

Table 1. Number of blocks modified to preserve the watermark (LENA, out of 4096 blocks)

qscale	0.8	0.4	0.2	0.1
blocks	0	1268	1561	2736

watermark has been added immediately after the last significant coefficient. It can be seen that the watermark is highly correlated with the host image. In active areas, the watermark is embedded in high frequency components, while it is embedded in low frequencies in smooth areas. This conforms to the masking effects of the human visual system.

Table 1 shows the number of blocks modified in the decoder to preserve the watermark. We used a quantization vector given in [6] multiplied with several values of *qscale*. With $qscale = 0.8$, the quantization is coarse enough to satisfy $q^c = q^w$. That no block has been modified with this quauntizer verifies the claim in Theorem 1. As this is not satisfied for other *qscale*'s, some blocks must be modified from initial decoded vectors, but it has been shown that after this modification, there was no watermark loss due to the integer rounding.

Fig. 3 shows how the watermark works. The image in (a) is a watermarked LENA with JPEG compression. If the correct key is used to verify the image, a repeated pattern is decoded as shown in (c). If a wrong key is used instead, we get a random pattern in (d), which indicates that the key is wrong. The image (b) is a modified version of image (a). The part which has been modified is detected using the correct key as shown in (e). In this experiment, K was set to 128 and the size of the group of blocks is 128×64 pixels. A group corresponds to a small unit pattern observed in (c).

6. CONCLUSION

We have given an algorithm to embed watermarks to authenticate JPEG images. The identity of images is strictly defined with regard to the quantization. We also discussed the survival of watermarks after JPEG decoding and have given a condition to prevent the loss of watermarks due to integer rounding in the decoding. The condition limits the fidelity of JPEG compression but by using different quantizers for watermarking and compression, the fidelity limitation can be removed. A specific decoding must be used in this case, which has been shown to work through the computer simulation.

7. REFERENCES

[1] F. Bartolini, et. al., "Image Authentication Techniques for Surveillance Applications", Proc. of IEEE, vol. 89, no. 10, pp. 1403-1418, Oct. 2001.

[2] P. W. Wong and N. Memon, "Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification," IEEE Trans. Image Processing, pp. 1593-1601, Oct. 2001.

[3] C. Y. Lin and C. F. Chang, "A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation," IEEE Trans. Image Processing, pp. 153-168, Feb. 2001.

[4] D. Kundur and D. Hatzinakos, "Digital Watermarking for Telltale Tamper Proofing and Authentication," Proc. of IEEE, vol. 87, no. 7, pp. 1167-1180, July 1999.

[5] H. Ito et. al., "A Decoding Scheme of Watermarked JPEG Images Based on Convex Projections", IPSJ Annual Convention, March 2004 (in Japanese).

[6] G. K. Wallace, "The JPEG Still Picture Compression Standard," IEEE Trans. on Consumer Electronics, vol. 38, no. 1, pp. 18-34, Feb. 1992.