A NOVEL REVERSIBLE WATERMARKING BASED ON AN INTEGER TRANSFORM

Shaowei Weng[1], *Yao Zhao*[1], *Jeng-Shyang Pan*[2] *and Rongrong Ni*[1]

[1]Institute of Information Science, Beijing Jiaotong University, Beijing 100444, P.R.China
[2]Department of Electronic Engineering, Kaohsiung University of Applied Sciences, Taiwan

ABSTRACT

A novel reversible data hiding scheme based on an integer transform is presented in this paper. The invertible integer transform exploits the correlations among four pixels in a quad. Data embedding is carried out by expanding the differences between one pixel and each of its three neighboring pixels. However, the high hiding capacity can not be achieved only by difference expansion, so the companding technique is introduced into the embedding process so as to further increase hiding capacity. A series of experiments are conducted to verify the feasibility and effectiveness of the proposed approach.

Index Terms— Reversible watermarking, Integer transform, Companding technique

1. INTRODUCTION

For some critical applications such as the fields of the law enforcement, medical and military image systems, it is crucial to restore the original image without any distortions after the watermark is extracted. The watermarking techniques satisfying these requirements are referred to as reversible watermarking.

The concept of reversible watermarking firstly appeared in the patent owned by Eastman Kodak [1]. After that, several reversible watermarking schemes were proposed in papers [2–8]. Xuan *et al.* [6] embedded data into the high-frequency wavelet coefficients by the companding technique [9] and used a preprocessing step to prevent the pixel overflow and underflow. Tian [7] expanded the value differences between two neighboring pixels to embed a bit without causing overflows or underflows. In Alatter's method [8], the differences between every two neighboring pixels in a quad were calculated according to a predefined order. Three bits were embedded into a quad by expanding the differences.

In the actual embedding processes in papers [7, 8], the differences less than a preset threshold were just expanded. Therefore, if a small integer value was selected as the threshold, the additional information, i.e., the location map recording the expanded positions, was hardly compressed into a short bitstream due to the nearly equal possibilities for expanded and non-expanded pixels, and accordingly, it almost consumed all the available hiding capacity. In general, the hiding capacity was very low at small thresholds.

To solve this problem, a novel reversible watermarking scheme combining an integer transform with the companding technique is proposed. The transform is proposed to calculate the difference between one pixel and each of its three neighboring pixels in a quad. The companding technique is introduced so that the differences larger than or equal to the threshold can also be expanded. Accordingly, the location map can be compressed into a very short bitstream to largely increase the embedding capacity.

The rest of the paper is organized as follows. In Section 2, the integer transform is proposed. A novel reversible watermarking scheme based on the transform is presented in Section 3. The performance analysis is discussed in Section 4. The experimental results are shown in Section 5 and finally we conclude the paper in Section 6.

2. AN INTEGER TRANSFORM

In a grayscale image I, every 2×2 adjacent pixels are grouped into a quad denoted by $q = \begin{bmatrix} u_0 & u_1 \\ u_2 & u_3 \end{bmatrix}, u_0, u_1, u_2, u_3 \in \mathbb{N}$.

The forward integer transform, $T(\cdot)$, is defined as

$$\begin{aligned} v_0 &= \lfloor \tfrac{u_0+u_1+u_2+u_3}{4} \rfloor \\ v_1 &= u_0 - u_1 \\ v_2 &= u_0 - u_2 \\ v_3 &= u_0 - u_3 \end{aligned} \tag{1}$$

The inverse integer transform $\begin{bmatrix} u_0 & u_1 \\ u_2 & u_3 \end{bmatrix} = T^{-1}\left(\begin{bmatrix} v_0 & v_1 \\ v_2 & v_3 \end{bmatrix}\right)$ is given by:

$$\begin{aligned} u_0 &= v_0 + \lceil \tfrac{v_1+v_2+v_3}{4} \rceil \\ u_1 &= u_0 - v_1 \\ u_2 &= u_0 - v_2 \\ u_3 &= u_0 - v_3 \end{aligned} \tag{2}$$

This work was supported in part by National Natural Science Foundation of China (No. 90604032, No. 60373028), Specialized Research Fund for the Doctoral Program of Higher Education, Program for New Century Excellent Talents in University, Specialized Research Foundation of BJTU, 973 program (No. 2006CB303104), Natural Science Foundation of Beijing (No. 4073038).

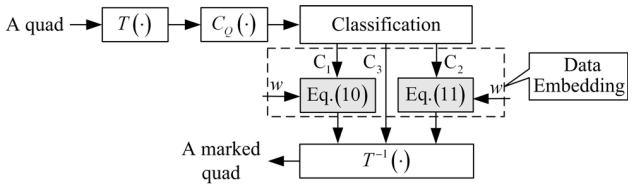

Fig. 1. Embedding process

In the following, the proof for the transform is given.

Proof: Substitute v_0, v_1, v_2 and v_3 from Eq. (1) into u_0 of Eq. (2), that is

$$\begin{aligned} u_0 &= v_0 + \lceil \tfrac{v_1+v_2+v_3}{4} \rceil \\ &= \lfloor \tfrac{u_0+u_1+u_2+u_3}{4} \rfloor + \lceil \tfrac{3u_0-u_1-u_2-u_3}{4} \rceil \\ &= \lfloor \tfrac{u_0+u_1+u_2+u_3}{4} \rfloor + \lceil u_0 - \tfrac{u_0+u_1+u_2+u_3}{4} \rceil \end{aligned} \tag{3}$$

Let $u_0 + u_1 + u_2 + u_3 = 4k + i$, where $0 \le i \le 3$ and $i, k \in \mathbb{Z}$, then we have

$$\begin{aligned} u_0 &= \lfloor \tfrac{4k+i}{4} \rfloor + \lceil u_0 - \tfrac{4k+i}{4} \rceil \\ &= \lfloor k + \tfrac{i}{4} \rfloor + \lceil u_0 - k - \tfrac{i}{4} \rceil \\ &= k + u_0 - k \\ &= u_0 \end{aligned} \tag{4}$$

The reversibility concerning u_1, u_2, and u_3 can be proved by simple mathematical manipulation of v_1, v_2, and v_3 in Eq. (2).

3. REVERSIBLE WATERMARKING BASED ON THE INTEGER TRANSFORM

In this section, a reversible watermarking scheme based on the transform is presented.

3.1. Embedding Process

The block diagram for the proposed reversible embedding scheme is illustrated in Fig. 1. For an 8-bit grayscale image, every 2×2 adjacent pixels are grouped into a quad in raster scan order. Note that all quads do not overlap each other. $T(\cdot)$ is applied to each quad to get its transformed quad. After that, the embedding process is performed.

This process is composed of three parts: 1) Companding: a quantized compression function, C_Q, is respectively applied to v_1, v_2 and v_3; 2) Classification: each quad is classified into one of three categories with the help of three outputs v_{Q1}, v_{Q2} and v_{Q3}; 3) Data embedding: it is implemented based on the quad's category. Here, three parts are respectively explained as follows.

3.1.1. Companding technique

Companding technique contains a compression function C and an expansion function E. For a signal x, C and E have relationship as follows:

$$E(C(x)) = x \tag{5}$$

In the digital case, C_Q and E_Q respectively represent the quantized versions of C and E, where Q denotes quantization function. The function C_Q adopted in this paper is given below.

$$\begin{aligned} x_Q &= C_Q(x) \\ &= \begin{cases} x & |x| < T_h \\ \mathbf{sign}(x) \times (\lfloor \frac{|x|-T_h}{2} \rfloor + T_h) & |x| \ge T_h \end{cases} \end{aligned} \tag{6}$$

where $\mathbf{sign}(\cdot)$ is the sign function, T_h is a preset threshold. In this paper, T_h is the same for all quads in an image. The expansion function is defined as

$$E_Q(x) = \begin{cases} x & |x| < T_h \\ \mathbf{sign}(x) \times (2|x| - T_h) & |x| \ge T_h \end{cases} \tag{7}$$

The companding error is calculated as follows.

$$r = |x| - |E_Q(C_Q(x))| \tag{8}$$

For $|x| < T_h$, the companding error r is zero. While for $|x| \ge T_h$, r is non-zero, and accordingly, it needs to be collected as a part of embedded data so as to losslessly restore the image.

3.1.2. Classification

Each quad q is classified into one of three categories: Class 1 ($\mathbf{C_1}$), Class 2 ($\mathbf{C_2}$), Class 3 ($\mathbf{C_3}$).

$\mathbf{C_1}$: For b_1, b_2 and $b_3 \in \{0, 1\}$, if the transformed quad $\begin{bmatrix} v_0 & \tilde{v}_1 \\ \tilde{v}_2 & \tilde{v}_3 \end{bmatrix}$ based on Eq. (10) satisfies Eq. (9), q is classified into $\mathbf{C_1}$.

$$\begin{aligned} &0 \le \tilde{u}_0 \le 255 \\ &0 \le \tilde{u}_0 - \tilde{v}_1 \le 255 \\ &0 \le \tilde{u}_0 - \tilde{v}_2 \le 255 \\ &0 \le \tilde{u}_0 - \tilde{v}_3 \le 255 \end{aligned} \tag{9}$$

where

$$\begin{aligned} \tilde{u}_0 &= v_0 + \lceil \tfrac{\tilde{v}_1+\tilde{v}_2+\tilde{v}_3}{4} \rceil \\ \tilde{v}_1 &= 2v_{Q1} + b_1 \\ \tilde{v}_2 &= 2v_{Q2} + b_2 \\ \tilde{v}_3 &= 2v_{Q3} + b_3 \end{aligned} \tag{10}$$

$\mathbf{C_2}$: For $q \notin \mathbf{C_1}$, if $\begin{bmatrix} v_0 & \tilde{v}_1 \\ \tilde{v}_2 & \tilde{v}_3 \end{bmatrix}$ given by Eq. (11) satisfies Eq. (9), q is considered to be in $\mathbf{C_2}$.

$$\begin{aligned} \tilde{v}_1 &= 2\lfloor \tfrac{v_1}{2} \rfloor + b_1 \\ \tilde{v}_2 &= 2\lfloor \tfrac{v_2}{2} \rfloor + b_2 \\ \tilde{v}_3 &= 2\lfloor \tfrac{v_3}{2} \rfloor + b_3 \end{aligned} \tag{11}$$

$\mathbf{C_3}$: The rest quads belong to $\mathbf{C_3}$.

Note that in the classification process, values 0 and 1 need to be assigned to b_i, $1 \leq i \leq 3$, respectively.

For each q in $\mathbf{C_1}$, if whichever of v_1, v_2 and v_3 is larger than or equal to T_h, it is classified in set $\mathbf{C}_{\geq T_h}$. Otherwise, it is considered to be in set $\mathbf{C}_{<T_h}$.

3.1.3. Data Embedding

The embedding process consists of two steps:

Step 1: Constructing to-be-embedded bitstream

A location map is created by assigning '1' to all the quads in $\mathbf{C_1}$ and '0' to the others. And then it is losslessly compressed by an arithmetic encoder to form a bitstream $\mathcal{L}$ with an unique EOS symbol at the end. Let L_S be the bit length of $\mathcal{L}$.

For each v in $\mathbf{C}_{\geq T_h}$, its companding error r given by Eq. (8) is collected into another bitstream $\mathcal{R}$. Thereby, there are four parts of information to be embedded: 1) bitstream $\mathcal{L}$; 2) bitstream $\mathcal{C}$ containing the LSBs of v_1, v_2 and v_3 of each quad in $\mathbf{C_2}$; 3) bitstream $\mathcal{R}$; 4) the payload. These four parts are combined together into a bitstream $\mathcal{W}$.

Step 2: Embedding the bitstream

For each q in $\mathbf{C_1}$, bits in $\mathcal{W}$ are embedded into v_1, v_2 and v_3 by left shifting and replacing the LSBs as described in Eq. (10). For each q in $\mathbf{C_2}$, bits in $\mathcal{W}$ are used to replace the LSBs of v_1, v_2 and v_3 as in Eq. (11). All the quads in $\mathbf{C_3}$ are kept unaltered. After all the bits in $\mathcal{W}$ are embedded, $T^{-1}(\cdot)$ is applied to get the marked image I_w.

3.2. Extraction Process

Every 2×2 adjacent pixels in the marked image I_w are grouped into a quad in the same way as in embedding. $T(\cdot)$ is applied to each quad to get its transformed one $q' = \begin{bmatrix} v_0 & \tilde{v}_1 \\ \tilde{v}_2 & \tilde{v}_3 \end{bmatrix}$.

The extraction process consists of two steps:

Step 1: Location map extraction

For each $q^{'}$, with $\tilde{v}_1$, $\tilde{v}_2$ and $\tilde{v}_3$ as inputs of Eq. (11), if the outputs satisfy Eq. (9), this quad is classified into set $\tilde{\mathbf{C}}_\mathbf{3}$. Otherwise, it belongs to set $\mathbf{C_3}$.

The LSBs of $\tilde{v}_1$, $\tilde{v}_2$ and $\tilde{v}_3$ in $\tilde{\mathbf{C}}_\mathbf{3}$ are collected into a bitstream $\mathcal{W}$. By identifying the EOS symbol in $\mathcal{W}$, the bits from the start until EOS are decompressed by an arithmetic decoder to retrieve the location map. For $q^{'}$ in $\tilde{\mathbf{C}}_\mathbf{3}$, if its location map value is 1, $q^{'}$ is classified into the set $\tilde{\mathbf{C}}_\mathbf{1}$. Otherwise, it belongs to set $\tilde{\mathbf{C}}_\mathbf{2}$. The bits after EOS in $\mathcal{W}$ are identified as bitstream $\mathcal{R}$, bitstream $\mathcal{C}$ and the payload.

Step 2: Restoration

For each $q^{'}$ in $\tilde{\mathbf{C}}_\mathbf{1}$, its original difference value v_1 is restored according to Eq. (12).

$$v_1 = \begin{cases} v_{Q1} & |v_{Q1}| < T_h \\ \text{sign}(v_{Q1}) \times (2|v_{Q1}| - T_h + w) & |v_{Q1}| \geq T_h \end{cases} \tag{12}$$

where $v_{Q1} = \lfloor \frac{\tilde{v}_1}{2} \rfloor$, $w \in \mathcal{R}$.

For $q^{'} \notin \tilde{\mathbf{C}}_\mathbf{1}$, its v_1 is retrieved as follows:

$$v_1 = \begin{cases} 2 \times \lfloor \frac{\tilde{v}_1}{2} \rfloor + w & \tilde{v}_1 \in \tilde{\mathbf{C}}_\mathbf{2} \\ \tilde{v}_1 & \tilde{v}_1 \in \mathbf{C_3} \end{cases} \tag{13}$$

where $w \in \mathcal{C}$.

The restoration of v_2, v_3 is similar to that of v_1. Finally, $T^{-1}(\cdot)$ is applied to reconstruct the original image I.

4. PERFORMANCE ANALYSIS

The difference modification caused by the companding technique and the data-hiding capacity are analyzed in this section.

4.1. Difference Modification

For each v in $\mathbf{C}_{\geq T_h}$, its marked value v_w is equal to $(2 \times v_Q + w)$, $w \in \{0, 1\}$. Let d be the difference between v_w and v.

If $v \geq T_h$, then we have the following equations

$$r = v - (2 \times v_Q - T_h) \tag{14a}$$

$$d = v_w - v = (T_h + w - r) \tag{14b}$$

If $v \leq -T_h$, then

$$r = -v - (-2 \times v_Q - T_h) \tag{15a}$$

$$d = v_w - v = (r + w - T_h) \tag{15b}$$

To conclude from above, when $v \in \mathbf{C}_{\geq T_h}$, each difference d is in the range of $[-T_h, T_h + 1]$. While for $v \in \mathbf{C}_{<T_h}$, d is in the interval $(-T_h, T_h + 1)$. In conclusion, each d would not exceed the range $[-T_h, T_h + 1]$.

4.2. Hiding Capacity

The maximum hiding capacity is given by:

$$\begin{aligned} D &= 3\|\mathbf{C_1}\| - L_S - \|R\| \\ &= \|\mathbf{C}_{\geq T_h}\| + \|\mathbf{C}_{<T_h}\| - L_S - \|R\| \\ &= \|\mathbf{C}_{<T_h}\| - L_S \end{aligned} \tag{16}$$

Where the operator $\|\cdot\|$ represents the length of a sequence or the cardinality of a set.

For each v in $\mathbf{C}_{\geq T_h}$, it can carry one bit, but meanwhile, it will introduce 1-bit companding error need to be embedded, so $\|\mathbf{C}_{\geq T_h}\| = \|R\|$. Thereby, it has no direct contribution to increase the hiding capacity. However, they are useful to enhance $\|\mathbf{C_1}\|$, and accordingly, the location map can be efficiently compressed into a short bitstream to get a small L_S. Finally, the data hiding is largely increased.

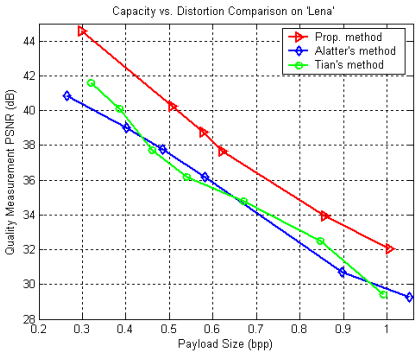

Fig. 2. Capacity vs. distortion comparison on 'Lena'

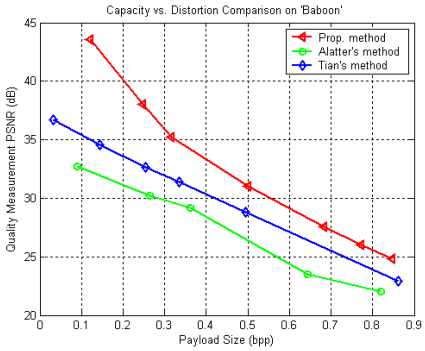

Fig. 3. Capacity vs. distortion comparison on 'Baboon'

5. EXPERIMENTAL RESULTS

For comparison, we experimented our method, Tian's method [7], and Alatter's method [8] on several commonly used images including 'Lena' and 'Baboon' with size 512×512. Our method and Alatter's can achieve embedding rates of close to 0.75bpp without multiple embedding. Multiple embedding was done while testing Tian's so as to achieve embedding rates above 0.5bpp.

The capacity vs. distortion comparisons among our method, Alatter's and Tian's are shown in Fig. 2 and Fig. 3.

As reported in Fig. 2, the top curve is the proposed method. By embedding a payload of the same size (bit length), the PSNR value of our method is about 1-3$\mathbf{dB}$ greater than that of the others. For example, the high embedding capacity of close to 0.3 bpp (bit per pixel) is achieved with above $44\mathbf{dB}$ in PSNR. However, at the same embedding rate, the PSNR values of the other two methods are both less than $42\mathbf{dB}$.

'Baboon' represents images with large areas of complex texture, so the experimental results are not good in Alattar's method and Tian's method. However, due to no pixels at both ends of its histogram image, most pixels can be still available for embedding at small T_h. For instance, in our method, the PSNR value above $44\mathbf{dB}$ can still be obtained with the embedding rate over 0.1bpp. Our method still outperforms Alattar's and Tian's at almost all PSNRs.

6. CONCLUSIONS

A novel reversible data hiding scheme based on an integer transform is presented in this paper. The companding technique is introduced so as to largely increase the number of quads in $\mathbf{C_1}$, and accordingly, the data-hiding capacity is increased especially at small thresholds. From the experimental results, our performance is superior to the others'.

7. REFERENCES

[1] C.W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, *Lossless Recovery of an Original Image Containing Embedded Data*, US patent:6278791, 2001.

[2] M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding for images," in *4th Information Hiding Workshop, LNCS*, New York, 2001, vol. 2137, pp. 27–41, Springer-Verlag.

[3] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, "Reversible data hiding," in *Proceedings of IEEE International Conference on Image Processing*, 2002, vol. 2, pp. 157–160.

[4] M.U. Celik, G. Sharma, and A.M. Tekalp, "Lossless watermarking for image authentication a new framework and an implementation," *IEEE Transaction on Image Processing*, vol. 15, pp. 1042–1049, 2006.

[5] M. Thodi and J.J. Rodrłguez, "Prediction-error based reversible watermarking," in *Proceedings of IEEE International Conference on Image Processing*, 2004, vol. 3, pp. 1549–1552.

[6] G.R. Xuan, C.Y. Yang, Y.Z. Zhen, and Y.Q. Shi, "Reversible data hiding using integer wavelet transform and companding technique," in *Proceedings of International Workshop on Digital Watermarking*, 2004.

[7] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, pp. 890–896, 2003.

[8] A.M. Alattar, "Reversible watermark using difference expansion of quads," in *Proceedings of IEEE Internation Conference on Acoustics, Speech, and Signal Processing*, 2004, vol. 3, pp. 377–380.

[9] M. Veen, F. Bruekers, A. Leest, and S. Cavi, "High capacity reversible watermarking for audio," in *Proceedings of the SPIE*, 2003, vol. 5020, pp. 1–11.