Reversible Watermarking Based on Histogram Shifting of Difference Image between Original and Predicted images

Su-Yeon Shin, Hyang-Mi Yoo, Jae-Won Suh
School of Electrical and Computer Engineering
Chungbuk National University
Cheongju, Korea
Email: ssy6061@naver.com, hmyoo82@cbnu.ac.kr, sjwon@cbnu.ac.kr

***Abstract*—Reversible watermarking is a technique that can recover an undistorted original image from a watermarked image. The proposed watermark embedding algorithm uses histogram shifting of the difference image between a modified original image and its predicted one. In the proposed algorithm, the predicted image that works well increases the embedding capacity, so that the reference pixels for prediction are adaptively selected and filtered and the other predicted pixels are directionally interpolated with the reference pixels. The simulation results demonstrate that the proposed algorithm generates good performances in the peak signal-to-noise ratio (PSNR) values and the embedding capacity.**

Keywords–Reversible watermarking; Histogram shifting; Predicted image; Reference pixel; Directional interpolation

I. INTRODUCTION

Illegal copies of digitized image can be easily and widely distributed through various communication channels and storage devices and be a serious problem for content owners. Watermarking technique can be a good solution to prevent the use of illegal contents. Watermarking technique can be categorized into three classes by the purpose: robust, fragile, and reversible watermarking. In the robust watermarking, the watermarked message must survive the various attacks such as resizing, cropping, filtering. For the fragile watermarking, the embedded watermark should be easily broken from the attacks. Reversible watermarking means that the original image and the watermark message can be completely recovered from the watermarked image without any distortion.

Reversible watermarking algorithms are studied many ways. The difference expansion scheme proposed by Tian [1] selected some expandable difference values of neighboring two pixels and embedded one bit into each of them. Ni *et al.* [2] found the maximum and the minimum pixel levels of the image histogram and shifted the histogram to embed the secrete data. Luo *et al.* [3] utilized the interpolation error, which is the difference between the interpolated pixel value and the corresponding pixel value. However, although these reversible watermarking algorithms based on histogram shifting make sufficient space for the watermark embedding, they suffer from the overflow and underflow problems because of the wrap around pixel levels caused by histogram shifting. Hong *et al.* [4] has extended Luo's work by generalizing the distribution of the reference pixels. However, if the distance between reference pixels become longer, the performance of Hong's algorithm get worse and worse. In this paper, we propose a new reversible watermarking algorithm tho overcome these problems.

The rest of this paper is organized as follows. Section II and Section III describe the watermark embedding and extracting procedures of the proposed algorithm, respectively. In Section IV, we demonstrate the effectiveness of the proposed algorithm. Finally, conclusions are drawn in Section V.

II. WATERMARK EMBEDDING PROCEDURE

The proposed reversible watermarking algorithm uses histogram shifting of the difference image between a modified original and its predicted images. Fig. 1 shows the proposed watermarking embedding procedure. A full explanation of each procedure is given below.

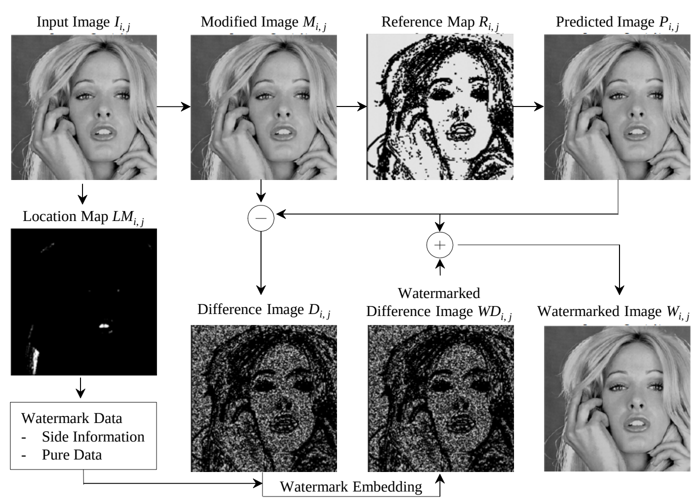


Figure 1. Watermark embedding procedure

A. Location Map and Modified Image

To overcome the wrap around problem, we need to monitor the lower bound pixel value "0" and the upper bound pixel value "255". Regarding the original image $I_{i,j}$ for $0 \leq i < M$ and $0 \leq i < N$, if the pixel value is equal to the lower bound pixel value or the upper bound pixel, we assign a "1" into the corresponding pixel location. Consequently, we obtain

 ISBN: 978-1-61208-353-7

an $M \times N$ binary image called as a location map. Next, we make a modified image $M_{i,j}$ by changing "0" into "1" and "255" into "254". Finally, the location map is losslessly compressed by using the joint bi-level image experts group (JBIG) compression algorithm and is inserted into some part of the embedded watermark data.

B. Predicted Image

The embedding capacity is proportional to the number of the most frequent pixel value at the difference image $D_{i,j}$, which can be obtained by obtaining a well predicted image. To do so, the predicted image $P_{i,j}$ is obtained by directional interpolation based on the proposed reference map $R_{i,j}$.

	0	1	2	3	4	5	6	7
0	0	1	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	0	1	1	1	1	0	1	1
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1

(a) If $\Delta \le 5$ ($\Delta = 5$)

$\overbrace{\hspace{3em}}^{\Delta}$

0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	1
0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	1

(b) If $\Delta > 5$ ($\Delta = 6$)

Figure 2. Reference map

1) Reference Map: As shown in Fig. 2, we define two different types of reference map $R_{i,j}$, where Δ is a pre-defined integer and the reference pixels are differently defined by Δ along the vertical and horizontal directions. If the spatial mesh interval for Δ gradually becoming wider, the prediction performance goes from bad to worse. To prevent the decrease of the prediction performance, if Δ is greater than 5, the reference pixels are located in a line. The positions associated with the reference pixels are notified by "0" and the others are "1", which is expressed in (1)

$$
\begin{aligned}
if \Delta \le 5: \quad R_{i,j} &= \begin{cases} 0 & \text{if } i\%\Delta = 0 \text{ and } j\%\Delta = 0 \\ 1 & \text{otherwise} \end{cases} \\
if \Delta > 5: \quad R_{i,j} &= \begin{cases} 0 & \text{if } i\%\Delta = 0 \text{ or } j\%\Delta = 0 \\ 1 & \text{otherwise} \end{cases}
\end{aligned} \tag{1}
$$

The reference pixels once selected should be preserved and the other pixels surrounded by the reference pixels are interpolated.

Next, we find the complex area and skip the prediction by using the reference map. It is very important because the complex area does not affect the embedding capacity and only causes the visual quality degradation of the watermarked image. To determine the complex area, we use the range function in (2). It returns the absolute difference between the maximum and minimum values of the given values.

$$
\begin{aligned}
&Range(x_1, x_2, x_3, x_4) = \\
&\qquad |Max(x_1, x_2, x_3, x_4) - Min(x_1, x_2, x_3, x_4)|
\end{aligned} \tag{2}
$$

If the given values are the equally spaced four corner reference pixel values and the return value of range function is greater than a pre-defined threshold T_0, the area surrounded by the four reference pixels are considered as a complex area. In this case, all pixels in the complex area are marked as reference pixels to prevent them from being interpolated in the process of making predicted image [4].

0	*1*	*1*	*1*	*1*	0
1	1	1	1	1	*1*
1	1	1	1	1	*1*
1	1	1	1	1	*1*
1	1	1	1	1	*1*
0	*1*	*1*	*1*	*1*	0

(a) $R_{i,j}$ for $\Delta \le 5$

40	***40***	***45***	***45***	***48***	50
45	47	49	50	52	***53***
50	50	53	53	53	***55***
52	55	60	65	70	***72***
55	60	63	65	70	***72***
60	***62***	***62***	***65***	***65***	70

(b) Corresponding $M_{i,j}$

40	***42***	***44***	***46***	***48***	50
44	47	49	50	52	***54***
48	50	53	53	53	***58***
52	55	60	65	70	***62***
56	60	63	65	70	***66***
60	***62***	***64***	***66***	***68***	70

(c) Interpolation result

Figure 3. Interpolation of boundary pixels

2) Pre-Processing of Boundary Pixels: As shown in Fig. 3, we calculate an imaginary boundary pixel values for smooth area by applying linear interpolation or low pass filtering to increase the prediction accuracy for the predicted image. In case of $\Delta \le 5$, the boundary pixel values between two reference pixels are linearly interpolated as shown in Fig. 3. In case of $\Delta > 5$, low pass filtering is applied to the reference pixels. The low pass filtered reference pixels are calculated by

$$
MR_{i,j} = \begin{cases} 1/4(M_{i-1,j} + 2 \times M_{i,j} + M_{i+1,j}) & \text{if } j\%\Delta = 0 \\ 1/4(M_{i,j-1} + 2 \times M_{i,j} + M_{i,j+1}) & \text{if } i\%\Delta = 0 \end{cases} \tag{3}
$$

The interpolated or low pass filtered boundary pixels are treated as reference pixels but are not reference pixels. They are only used for prediction mode decision and directional interpolation to make a predicted image.

40	42	44	46	48	50
44	47	49	50	52	54
48	50	53	53	53	58
52	55	60	65	70	62
56	60	63	65	70	66
60	62	64	66	68	70

(a) Diagonal

40	42	44	46	48	50
44	47	49	50	52	54
48	50	53	53	53	58
52	55	60	65	70	62
56	60	63	65	70	66
60	62	64	66	68	70

(b) Vertical

40	42	44	46	48	50
44	47	49	50	52	54
48	50	53	53	53	58
52	55	60	65	70	62
56	60	63	65	70	66
60	62	64	66	68	70

(c) Horizontal

40	42	44	46	48	50
44	47	49	50	52	54
48	50	53	53	53	58
52	55	60	65	70	62
56	60	63	65	70	66
60	62	64	66	68	70

(d) Backward Diagonal

Figure 4. Prediction mode

3) Prediction Mode and Directional Interpolation: As shown in Fig. 4, the pre-processed boundary pixels are used for prediction mode decision. There are five prediction modes: diagonal mode, vertical mode, horizontal mode, backward diagonal mode, and plane mode. The mode having the smallest mean sum of absolute difference (MSAD) between far away shaded pixels is determined as the prediction mode of the

 ISBN: 978-1-61208-353-7

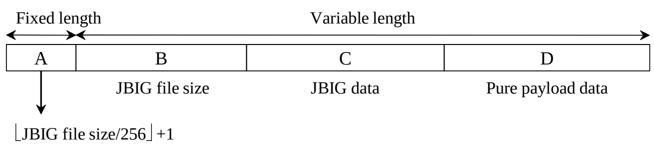
current block. The MSAD is given by (4).

$$MSAD = \frac{1}{n}\sum_{l=1}^{n}|x_l - y_l| \tag{4}$$

For examples of the above Fig. 4, the MSADs for diagonal mode, vertical mode, horizontal mode, and backward diagonal mode are 26, 20, 10, and 9, respectively. Therefore, the prediction mode of the current block is the backward diagonal mode. Therefore, the pixels of the current block are directionally interpolated by the backward diagonal mode. However, if the MSAD is larger than pre-defined threshold T_1, the block is decided as a plane mode and bilinear interpolated.

C. Data Structure of the Watermark Message

In order to be able to recover the original image and the watermark message from a watermarked image, the embedded watermark data have to be designed considering the extraction rule. The data structure of the embedded message is shown in Fig. 5 [5].



- "A" field: An eight bit integer indicating how many next bytes are used for notifying the length of the compressed location map.
- "B" field: Representing the real file size of the compressed location map by JBIG.
- "C" field: Compressed bitstream of the location map.
- "D" field: Pure watermark data.

Figure 5. Data structure of the watermark message

D. Watermark Embedding by Histogram Shifting

The first thing for watermark embedding is making a histogram of difference image $D_{i,j}$. The difference image $D_{i,j}$ is obtained by subtracting the predicted image $P_{i,j}$ from the modified image $M_{i,j}$ for $0 \leq i < M$ and $0 \leq j < N$. Fig. 6(a) shows the histogram of the difference image.

Secondly, the histogram shifting is performed around the first and the second maximum peak point (PP) pixel values in order to make the data embedding space. "PP_1" and "PP_2" are the pixel values that have the first and second highest number of pixels in ascending order. As shown in Fig. 6(b), we empty the "$PP_1 - 1$" and "$PP_2 + 1$" levels using bi-directional histogram shifting. It means that if the pixel value of $D_{i,j}$ is less than or equal to "$PP_1 - 1$", the corresponding pixel value is decremented by "1" throughout the whole image. If the pixel value of $D_{i,j}$ is greater than or equal to "$PP_2 + 1$", the corresponding pixel value is incremented by "1".

Therefore, the shifted difference image $SD_{i,j}$ can be expressed as

$$SD_{i,j} = \begin{cases} D_{i,j} - 1 & \text{if } D_{i,j} \leq PP_1 - 1 \\ D_{i,j} + 1 & \text{if } D_{i,j} \geq PP_2 + 1 \end{cases} \tag{5}$$

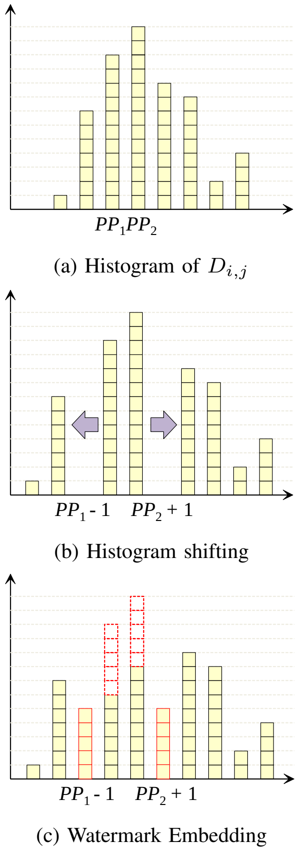


Figure 6. Structure of the watermark data

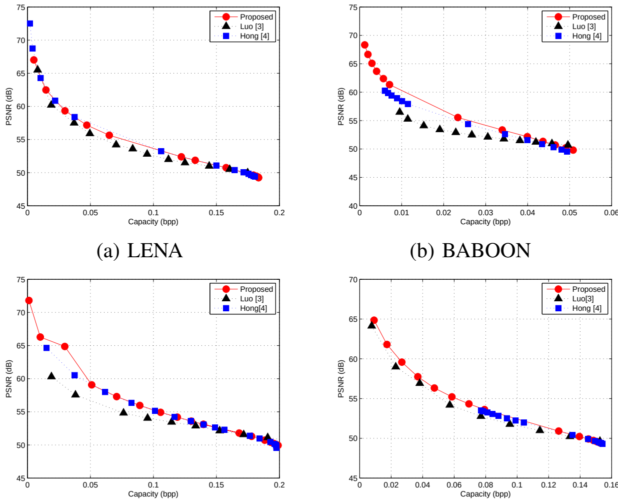
Next, the watermark message $W(k)$ is embedded into the two PP values. The shifted difference image is scanned again using its raster scan order. Once the "PP_1" are "PP_2" values are encountered, we sequentially check the watermark message $W(k)$. If the checked bit is a "1", the pixel value "PP_1" or "PP_2" is changed into "$PP_1 - 1$" or "$PP_2 + 1$", respectively. If the checked bit is a "0", there is no change as shown in Fig. 6(c). As a result, we can get the watermarked difference image $WD_{i,j}$ and finally obtain the watermarked image $W_{i,j}$ by adding with the predicted image $P_{i,j}$.

III. PROPOSED WATERMARK EXTRACTION

If the receiver has the watermarked image and the key information for extraction, the watermarked image can be separated into the original image and the watermarked data. The key information for watermark extraction are pre-determined T_0 and T_1, Δ, PP_1 and PP_2.

First, we make the predicted image $P_{i,j}$ by using Δ, T_0 and T_1. Because the reference pixels are not changed during the embedding process, the predicted image is exactly the same as that obtained in the embedding procedure.

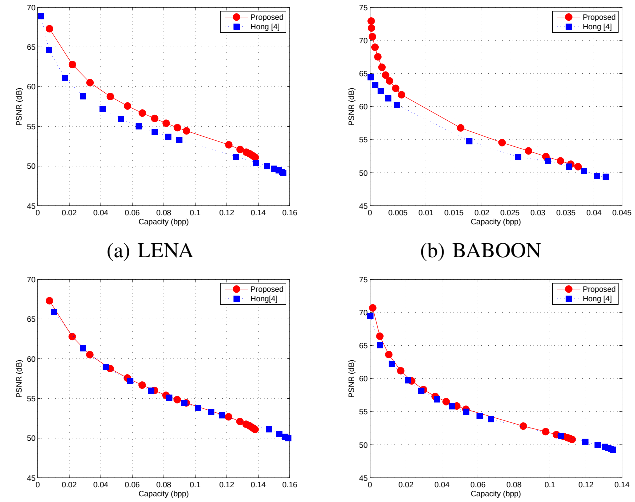
Second, we generate difference image $D_{i,j}$ between the watermarked image $W_{i,j}$ and the predicted image $P_{i,j}$. During the scanning the difference image $D_{i,j}$, we can extract the embedded watermark data $W(k)$ using the PP_1 and PP_2.



(a) LENA (b) BABOON

(c) AIRPLANE (d) PEPPER

Figure 7. Results for $\Delta = 3$



(a) LENA (b) BABOON

(c) AIRPLANE (d) PEPPER

Figure 8. Results for $\Delta = 6$

Whenever the corresponding pixel level is equal to "PP_1" or "PP_2", the extracted bit is "0". If the corresponding value is "$PP_1 - 1$" or "$PP_2 + 1$", the extracted bit "1".

Third, we re-scan the entire difference image $D_{i,j}$ and recover the histogram of the difference image. If the corresponding value is less than or equal to "$PP_1 - 1$", "1" is added and if the corresponding value is greater than or equal to "$PP_2 + 1$", "1" is subtracted. The returned difference image generates the modified image $M_{i,j}$ by adding the predicted image $P_{i,j}$.

Next, we parse the compressed location map information from the extracted watermark message $W(k)$. Finally, we recover the original image $I_{i,j}$ by using the modified image $M_{i,j}$ and the decompressed location map data.

IV. Experimental Results

In order to evaluate the performance of the proposed reversible watermarking algorithm, we performed computer simulations on typical 512×512 8-bit images: LENA, BABOON, AIRPLANE, and PEPPER. The performance of the proposed reversible watermarking algorithm has been compared to those presented by Luo *et al.* [3] and Hong *et al.* [4] in terms of the embedding capacity versus the PSNR of the watermarked image. In Fig. 7 and Fig. 8, we just showed two simulation results: $\Delta = 3$ and $\Delta = 6$. The simulation conditions are like this: various T_1 (1, 2, ..., 10, 20, ...80) and fixed $T_2 = 10$.

In both results, we can find the common property that the small T_1 generates high PSNR value but low embedding capacity. In Fig. 7, the proposed algorithm achieves slightly better embedding efficiency than that of the Hong's algorithm. The superiority is from the low pass filtering of the reference pixels and directional interpolation according to the prediction mode in the proposed algorithm. The excellence of the proposed algorithm is clearly shown in the case of $\Delta = 6$. By adapting the new structure of the reference pixels for prediction, the prediction errors are reduced and then it increases the embedding capacity.

V. Conclusions

In this paper, we have proposed a reversible watermarking algorithm based on the histogram shifting of the difference image between a original image and a predicted image. In order to solve the underflow and overflow problems, a location map is generated, compressed, and embedded as a part of the watermark message. In previous works, the reference pixels are distributed by equally spaced interval and the pixels surrounded the reference pixels are just bi-linearly interpolated. To enlarge the embedding capacity while keeping the visual quality of the watermarked image, we have suggested the alternative way to make reference pixel map and predicted the smooth area by directional interpolation. From the simulation results, we can conclude that the the proposed algorithm generates good watermarked image quality and embedding capacity.

References

[1] J. Tian, "Reversible Watermarking by Difference Expansion," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 8, Aug. 2003, pp. 890-896.

[2] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 3, Mar. 2006, pp. 354-362.

[3] L. Luo, Z. Chen, M. Chen, and X. Zeng, "Reversible Image Watermarking Using Interpolation Technique," *IEEE Trans. on Information Forensics and Security*, vol. 5, no. 1, Mar. 2010, pp. 187-193.

[4] W. Hong, and T. Chen, "Reversible Data Embedding for High Quality Images Using Interpolation and Reference Pixel Distribution Mechanism," *Journal of Visual Communication and Image Representation*, vol. 22, no. 2, Feb. 2011, pp. 131-140, 2011.

[5] H.M. Yoo, S.K. Lee, and J.W. Suh, "High capacity reversible data hiding using the histogram modification of block image," in *Proc. ISCAS*, May 2010, pp. 1137-1140.

 ISBN: 978-1-61208-353-7