Adaptive Thresholding of an Efficient Generalized Integer Transform Based Reversible Watermarking

Ka-Cheng Choi and Chi-Man Pun

Faculty of Science and Technology, University of Macau, Macau,

China {mb05425, cmpun}@umac.mo

Abstract. In this paper, a generalized integer transform based reversible watermarking algorithm is extended. In the original method, the threshold which affects the embedding capacity needs to be preselected, if an unsuitable threshold is selected, there may not be enough space for the watermark or may even degrade the visual quality. With the proposed method, suitable threshold is selected adaptively, which ensures enough embedding capacity while keeps the distortion low. Moreover, the proposed method has more capacity available for embedding as it introduces lesser overhead information through further reducing the location map. Overall, it provides more embedding capacity whereas improves the visual quality of the embedded image.

Keywords: generalized integer transform, location map, reversible watermarking, threshold

1 Introduction

Recently, many researches have been carried out on the topic of reversible watermarking; a type of watermarking that allows total recovery of the original image without any distortion after the hidden data is retrieved from the marked content. The characteristic that the image can be restored is desirable, especially in the fields of law enforcement, medical and military image systems where even a small distortion is intolerable. Many valuable reversible watermarking algorithms have been published up to today, which can be classified into three categories [1]: data compression [2, 3], difference expansion [4-6] and histogram modification [7, 8]. Data compression is to losslessly compress the image to be overlaid to leave space for embedding watermark bits. It usually involves complex computation and has limited capacity [9]. Difference expansion is first proposed by Tian [4], many variants are then presented [5, 10]. It is an integer wavelet transform where the watermark bits are hidden in the expanded differences. It usually achieves a high embedding capacity and keeps distortion low [11], that's why it attracts a lot of attention including us. Among the variants of Tian's DE, Alattar [5] started extending it in a generalized manner by applying it to a triple or quad of pixels which increases the hiding ability and computation efficiency. Histogram modification usually utilizes the zero or minimum points of the image histogram and modifies the pixel values to leave space for bits to be embedded, but its embedding capacity is usually not as high as difference expansion [12].

In this paper, a recently proposed generalized integer transform reversible watermarking method [13] which based on Tian's DE is extended to improve the weakness that an influential threshold value need to be preselected. In the proposed method, given the watermark, a suitable threshold value is determined to achieve the target that whole watermark can be embedded with the least visual degradation. Moreover, overhead information has been reduced in size, so more space is left for watermark. In the next two sections, the proposed method will be described in detail, for better illustration, 0 and 0 show the flowchart of the embedding and extraction procedure of the proposed method for reversible watermarking respectively.

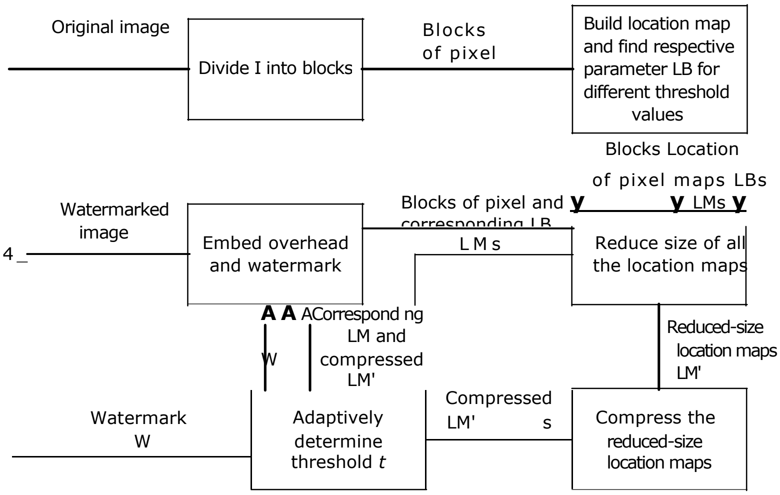

Fig. 1. Flowchart of embedding procedure for reversible watermarking

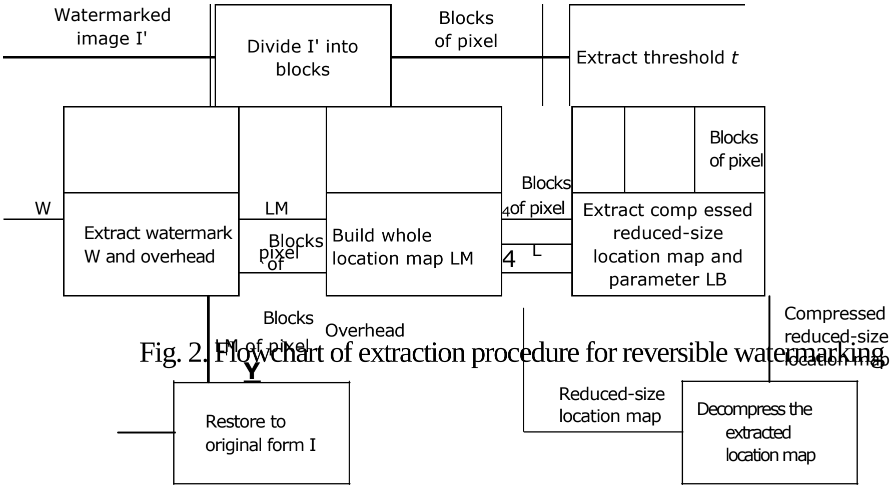

Fig. 2. Flowchart of extraction procedure for reversible watermarking

2 Embedding Procedure of Adaptive Thresholding for Reversible Watermarking

As mentioned before, correctly choosing the threshold t is crucial, since it may helps to raise PSNR of the watermarked image I'. In order to do this, at first, the number of bits can be embedded for different value of t need to be known, so that enough space for the watermark can be ensured. Since t affects whether the blocks belong to embeddable group through the requirement that embeddable blocks need to have $v(x) < t$, therefore, the larger t is, the more embeddable blocks are available for embedding, and more bits can be hidden in the image. However, as explained in [13], the distortion introduced to the embeddable block is close to its variance v(x), so the value of t should not be too large so that the visual quality is guaranteed. To compromise between the embedding capacity and visual quality, the smallest t that can accommodate the given watermark W should be chosen.

In fact, the embedding capacity of an image is the number of embeddable blocks times $(n - 1)$ bits, where n is the size of the blocks, then deduct the overhead which is mainly consumed by the location map. Although the size of the overhead other than the location map is fixed, the size of location map fluctuates a lot, therefore, every location map need to be built and compressed in order to find its length for different t. Now, with the proposed method, given the watermark, according to its length, a suitable threshold t that can accommodate the whole watermark while keeps distortion low can be chosen. With experiments conducted, we found out that threshold t should not be larger than 100, otherwise the PSNR value drops below 30, therefore, the range of t in our experiments are kept between 0 and 100.

3 Extraction Procedure for Reversible Watermarking

At first, I' is divided into blocks in the same way as embedding and the threshold t is extracted which is needed to distinguish E U C blocks from 0 blocks, the way that it is extracted is to get the LSBs of the first 7 pixels of the image. With t, location map L that distinguishes E u C and 0 can be built, followed by, we need to extract the original LSBs of those 7 pixels (B) in order to restore those blocks to the state that the compressed LM' is just embedded into them. As those bits are embedded in the E U C blocks starting from the end, therefore, what have to do is continue to extract the first $n - 1$ LSBs of E U C blocks starting from the end which can be known from L until that 7 LSBs are got. Notice that, here the first bit of the extracted LSBs of every block is its state, indicates that it is embeddable or changeable, which is then used to modify the location map L to L' by replacing with the respective states got from previous step. The 7 LSBs extracted just now are then used to restore the LSBs of the first 7 pixels.

Now, extraction of the compressed LM' and related information (A) can be done, the way that it is extracted is the same as [13], the difference is that a few bits extracted is used to store the overhead including *LB.* Compressed LM' is first uncompressed, then with LM', *LB* and L', the whole finalized location map LM that identifies the embeddable blocks can be built. To change L' to LM, i.e. location map that identifies E U C to only identifies E, the only thing need to do is to check the state of every E U C blocks to see if it is embeddable. This is done with the help of *LB* and LM', for every E U C blocks that with $v_h(x) > LB$, where x is a block vector of I', change its state as LM', 1 indicates embeddable, 0 indicates changeable, the states of the rest of the block remain unchanged, since 0 blocks will not be embeddable, so keeps as 0, and E U C blocks with $v_h(x) < LB$ must be embeddable, so keeps as 1.

Then, extract the watermark portion (D) in the same way as mentioned in [13], CLSBI and $_{CLSB2}$ are extracted altogether from D, and the block index of the end of portion D is recorded. Finally, what is left is to restore the image to its original form as described in [13] until it reaches the block index recorded before, as the blocks afterward are unchanged, so nothing have to be done, except the blocks that are used to embed FLSB (B) which is needed to be recovered, it is recovered using the same way as recovering the blocks that are used to embed the compressed LM', for changeable blocks, the LSBs used in recovering are stored in $_{CLSB2}$ which is attached at end of W. Therefore, after recovery, the attached LSBs CLSBI and CLSB2 will be removed, and the extracted portion D will leave over with the watermark only which is what is needed.

4 Experimental Results

As explained in [13], 4 x 4 *(n* = 16) is the size of the blocks that gets the best performance, therefore, 4 x 4 sized blocks will be used in the experiments conducted.

In the proposed method, the size of the location map has further been reduced; Fig. 3 shows the location map size of the original method and the proposed method of "Airplane" for comparison, as we can see, the size reduced is quite significant and satisfactory. The location map size is reduced obviously and dramatically from t = 10 onwards, i.e. when the data to be embedded is not too small, the effect of this modification is quite effective. This reduction of the overhead provides more capacity for embedding the payload, therefore, provides larger embedding capacity with the same PSNR value.

Followed by, the performance of the original method, proposed method and other three algorithms• 1) Tian's method, 2) Alattar's method and 3) Weng's method are compared, in Fig.4, the embedding capacity versus image visual quality data for these methods of "Airplane" are shown, for the original method, the preselected threshold is set to 70. From the figure, we can see that our method is better than all the other four methods.

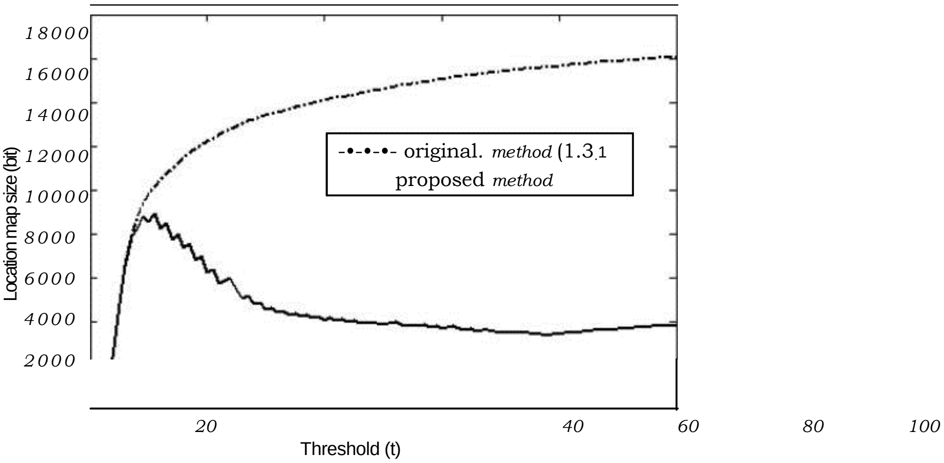

Fig. 3. Comparison of location map size between the original and the proposed modified method for "Airplane"

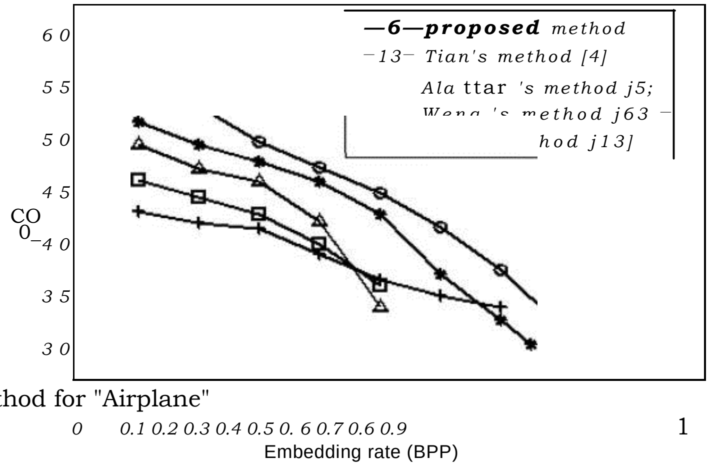

Fig. 4. Performance comparison of our method with the other three methods for "Airplane"

(a) **(b)** (c)

Fig. 5. Embedded "Airplane" images, (a) 0.5 bpp (43.84 dB), (b) 1.0 bpp (31.03 dB) and (c) 1.95 bpp (24.53 dB)

Finally, the embedded "Airplane" images by the proposed method with payloads of 131 072 bits, 262 144 bits, and 512 000 bits are shown in Fig. 5(a), (b) and (c) respectively. In Fig. 5(c), with a high embedding capacity of 1.95 bpp, the image still has 24.53 dB and its image content is preserved quite well.

5 Conclusion

In this paper, a recently proposed generalized integer transform reversible watermarking is developed in two aspects. One is an improved distortion control through adaptively chosen threshold during iterative step in the algorithm. The other is further reducing the size of the location map to be embedded, thus less capacity is used for storing overhead so as to increase its embedding capacity. Experimental results also indicate that the proposed method has improved performance compared with state of the art existing method.

References

1. R. Agrawal and R. Srikant, "Privacy-preserving data mining," *SIGMOD Rec.*, vol. 29, pp. 439-450, 2000.
2. M. U. Celik, *et al.*, "Lossless generalized-LSB data embedding," *Image Processing, IEEE Transactions on*, vol. 14, pp. 253-266, 2005.
3. L. Kamstra and H. J. A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *Image Processing, IEEE Transactions on*, vol. 14, pp. 2082-2090, 2005.
4. T. Jun, "Reversible data embedding using a difference expansion," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 890-896, 2003.
5. A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *Image Processing, IEEE Transactions on*, vol. 13, pp. 1147-1156, 2004.
6. W. Shaowei, *et al.*, "Reversible Watermarking Based on Invariability and Adjustment on Pixel Pairs," *Signal Processing Letters, IEEE, vol.* 15, pp. 721-724, 2008.
7. N. Zhicheng, *et al.*, "Reversible data hiding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, pp. 354-362, 2006.
8. T. Wei-Liang, *et al.*, "Reversible Data Hiding Based on Histogram Modification of Pixel Differences," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, pp. 906-910, 2009.
9. L. Lixin, *et al.*, "Reversible Image Watermarking Using Interpolation Technique," *Information Forensics and Security, IEEE Transactions on, vol.* 5, pp. 187-193, 2010.
10. D. M. Thodi and J. J. Rodriguez, "Expansion Embedding Techniques for Reversible Watermarking," *Image Processing, IEEE Transactions on*, vol. 16, pp. 721-730, 2007.
11. Y. Q. Shi, *et al.*, "Lossless data hiding: fundamentals, algorithms and applications," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, 2004, pp. 11-33-6 Vol.2.
12. Z. Bin, *et al.*, "A near reversible image watermarking algorithm," in *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, 2010, pp. 2824-2828.
13. W. Xiang, *et al.*, "Efficient Generalized Integer Transform for Reversible Watermarking," *Signal Processing Letters, IEEE*, vol. 17, pp. 567-570, 2010.