Image Tamper Detection and Recovery based on Dilation and Chaotic Mixing

Hao-Chun Wang¹, Wei-Ming Chen²*, Ping-Yi Lee²

¹Department of Computer Science and Information Engineering, National Taipei University of Technology, Taiwan
²Department of Computer Science and Information Engineering, National Ilan University, Taiwan

Abstract In this paper, we propose an efficient method for image tamper detection and recovery. We separate the image into several blocks, and share a block information to two other blocks. It means, there have two copies of one block information for each non-overlapping block. In implementing, we improve Lee's algorithm of watermark embedding, especially when the tampered area is a rounded region or a text field, and our scheme has higher PSNR value than Lee's method. During image recovery, we use the image inpainting approach to fill the tampered region of whole image. The experimental result shows that our scheme is effective than Lee's interpolation method.

Keywords Tamper Detection, Recovery, Digital Watermark

1. Introduction

In the last decade, there has been a rapid growth in using digital image data. People can easily modify a copyrighted image data by an image-processing program. For this reason, how to efficiently and accurately detection of tampered region has more focus on this research.

The above condition, a great number of research focus on the method of tamper detection, [1-2] use dual watermark approach to embed a watermark, and we use those research for our main process. In [3], it proposed a method that uses different number of space to embed non-smooth and smooth blocks. Another different way in image recovery that uses hierarchical image tamper detection, it proposes in [4]. In [5-7], they describe the theory of tamper detection. A concept of demosaicing artifacts for watermarking is discussed in [8].

In our proposed method, when the process needs to modify the last significant bit (LSB), we regulate some bit of the most significant bit (MSB). The above step that could make the higher PSNR value for an embedding image. In the recovery process, we use the Dilation algorithm to define the tampered region as invalid blocks. This approach replaces the general hierarchical detection [4], and effective to detect tampered from a rounded region or a text field. More theory and concept of Dilation algorithm is discussed in [9-10]. In the other hand, if a large tampered region has coming and could not be recovered by watermark. In this case, we use the texture synthesis method to overcome this problem. Above process is called image inpainting, and discussion of the image inpainting based on the texture synthesis is in [11-14]. In summary, we improve the method that is used to detect a rounded region or a text field, and make whole process of detection became faster than previous method.

The rest of paper is organized as follows: Section 2 described the proposed scheme; Section 3 shows the experimental results; Section 4 summarizes this paper.

2. The Proposed Method

In this section, we will explain the detail of proposed method for each step in the flow chart as shown in Figure 1. In (a), before the Watermark Embedding process, we adjust some bit to reduce the gray-level value difference between the original value and modified value, and repeat the above step at after the Watermark Embedding process for the same purpose. In (b), we provided of tamper detection method that described as follows: First, we check the last two bit of 12-bit watermark for each block that we embedded from previous flowchart (a), and marks those fault blocks as the invalid region. When the all blocks were checked, and wait to recover it at next step. In the part of Tamper Recovery process, we recover the invalid region from the watermark copies, if the watermark copies were broken, then we use the image inpainting algorithm to recover image.
Image Tamper Detection and Recovery based on Dilation and Chaotic Mixing

2.1.1 Preparation

The size of all original image which we used in this research is 512×512 pixels size, and separate it into blocks of 2×2 pixels size. In Fig 2 (a), we mark all index of these blocks in the original table, and set they in random position by chaotic mixing algorithm as shown in formula (1).

\[ X' = (k \cdot X \mod N) + 1 \]  

In formula (1), where \( k \) is a prime, \( N \) is a size of total blocks, \( X \) as an index of block that we singed in original table, \( X' \) is a new index by through the Chaotic Mixing algorithm.

After the process of Chaotic Mixing, we could obtain a LUT as shown in Fig. 2.

2.1.2 Watermark Generation

In this part, we generate the watermark and checking bit for each block. First of all, we separate the LUT into two major parts as upper half and lower half. Then, we pick up one block from each part, and give them a partner relationship, (e.g., in Fig. 3, block A and block A' are partner). Now, we can go to the next step to calculate the average value for a block and its partner.

In Fig. 4, the block A and its partner block A' are ready to generate 12-bit watermark, we extract the top five bits as MSB to generate the watermark. In formula (2) and (3), a protection code \( p \) is a result from XOR calculating, a verification code \( v \) is computed using NOT operator.

\[ p = a_7 \oplus a_6 \oplus a_5 \oplus a_4 \oplus a_3 \oplus a' \oplus a' \oplus a' \oplus a' \oplus a' \oplus a' \oplus a' \]

\[ v = \neg p \]

2.1.3. Average Value Correction

Before watermark embedding, we have to briefly adjust the average value of block A and block A'. When we generate the 12-bit watermark in above Fig. 4, the last three bits LSB are useless bits. Thus, it has a problem of the difference between the true value and MSB value. For this reason, we correct the last bit of the MSB, forcing the result has similar original value. This method is also suitable to the block value correction step when after the block was embedded 12-bit watermark.

2.1.4. Watermark Embedding

In this part, we have to embed the 12-bit watermark that we obtain in watermark generation step. In Fig. 5, it illustrates the 12-bit watermark how to embed in the LSB of block A and its partner the block A'. To embedding the same watermark in the two different blocks is helpful for recovery, when the image was tampered, we can use the partner relationship to recover another one block.
In the end of this process, we must adjust the bit of pixels at the two blocks A and A’. Assumed that one original pixel of block A is 10001000₂ that equal 136₁₀, and the embedded three bit of 12-bit watermark is 111₂. If we embed it directly, it is 10001111₂ that equal 143₁₀, the difference between two values is 7. For this reason, we adjust the last bit of the MSB as 10000111₂ that equal 135₁₀, now, the difference between two values is 1. Thus, this step could really reduce the difference between two values and improve the PSNR value.

2.2.1. Tamper Detection

In the image tamper detection approach, the proposed method is different with Lee’s algorithm [1]. Our algorithm has more efficient and correct in tampered region detection. In [1] algorithm, it has complex 3-level hierarchical steps for detecting tampered region. But, in our algorithm, we only use the level-1 step of those complex steps. In Fig. 6, when the block was marked invalid, we use the dilation algorithm for the invalid regions, and define the result as a new invalid region. The experiment proved that the proposed method is better than [1], especially when the tampered area is a rounded region or a text field.

2.2.2. Recovery and Texture Synthesis

When all of the invalid blocks were find, we are ready to recover image. An invalid block has two approaches to recover itself: one is using the 12-bit watermark from the partner block; another way is using image inpainting. Only the partner block was marked invalid, otherwise we use the 12-bit watermark to recovery. In image inpainting, we use the texture synthesis and the priority for filling image, it has more discussion and proof in [11].

![Figure 5](image-url) Embedding location of block A and block A’.}

![Figure 6](image-url) (a) The tampered image. (b) The invalid regions. (c) The new invalid regions by our proposed.

![Figure 7](image-url) The recovered image from difference tampered. (A) 65% tampered. (B) 70% tampered. (C) 75% tampered. (D) 80% tampered. (E) 85% tampered. (F) 90% tampered. (G) 95% tampered. (H) 97% tampered.
3. Experimental Results

At the first experiment as follows, we evaluate the size of tampered region and its recovery ability. According to the result, we could see the Lena’s contour by after recovering 97% tampered region, and the PSNR is better than other algorithm.

In the second experiment, we use the different shapes to cover the image that is shown in Figure 8 and Table 2. In the next experiment, we use the irregular shape to cover image as cutting image. The experimental result is shown in Figure 9 and Table 3.

![Figure 8. The tampered region by different shapes.](image)

| Table 2. The PSNR value from the above Fig. 8. |
|-----------------------------------------------|
| Image | Location | Tampered | PSNR |
| Lena  | Center   | 9%       | 38.85 |
| Lena  | Left     | 50%      | 30.04 |
| Lena  | Left,Top | 50%      | 30.17 |

![Figure 9. The tampered region by irregular shape.](image)

| Table 3. The PSNR value from the above Fig. 9. |
|-----------------------------------------------|
| Embedded Image | Tampered | PSNR |
| Lena           | 46%     | 31.8 |
| Lena           | 54%     | 28.43 |
| Baboon         | 20%     | 31.58 |
| Baboon         | 80%     | 20.53 |

In the last experiment, we use a text field as a tampered region. Because of dilation algorithm, we could still have great PSNR value in the text-field attack. The experimental result is shown in Figure 10 and Table 4.
4. Conclusions

In this paper, we proposed an efficient method for image tamper detection and recovery. According to the result of experiment, our method has resilience for cutting and modification attacks, especially when the tampered area is a rounded region or a text field. In process of watermarking, our scheme has higher PSNR value than Lee's method. During the process of the tamper detection, we only use the verification code 'v' of the 12-bit watermark, and do not need the original image. Avoiding errors on the rounded region or the text region, when the all blocks of image were checked, we dilate the invalid regions. In the image recovery phase, our method is to use the image inpainting algorithm. Image inpainting is very different with traditional Lagrange Interpolation algorithm, it includes texture synthesis and the priority. The experimental result proofs that using the image inpainting algorithm is better than interpolation algorithm, whether in terms of intuition or PSNR value.

Our approach is superior in two aspects: The LSB embedded 12-bit watermark is less sensitive to original image, and we can keep most information of image; we use the image inpainting technique to recover image. It keeps the main structure of image, especially when the tampered region is large. In summary of above the advantages, our proposed method that is effective for tamper detection and recovery.

### Acknowledgements

The authors would like to thank the National Science Council, Taiwan, R.O.C. for financially supporting this research under Contract MOST 103-222 -E-197-014-.

### REFERENCES

[1] Tien-You Lee and Shinfeng D. Lin, “Dual watermark for image tamper detection and recovery,” Pattern Recognition, vol. 41, no. 11, pp. 3497-3506.

[2] Chu-Hsing Lin, Jung-Chun Liu, Wei-Cheng Hsu, Hsing-Weng Wang, Wei-Chih Lin, and Jian-Wei Li, “Image Tampering Detection and Recovery Using Dual Watermarks and Cyclic Redundancy Checks,” in Advanced Communication and Networking, Ed. Chang Chin-Chen, Vasilakos Thanos, Das Purmundu, Kim Tai-hoon, Kang Byeong-Ho and Khurrum Khan Muhammad, Springer Berlin Heidelberg, 2010, pp. 134-143.

[3] Ching-Sheng Hsu and Shu-Fen Tu, “Image tamper detection and recovery using differential embedding strategy,” in Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference, pp. 399-402, Aug. 2011.

[4] Phen Lan Lin, Chung-Kai Hsieh and Po-Wei Huang, “A hierarchical digital watermarking method for image tamper detection and recovery,” Pattern Recognition, vol. 38, no. 12, pp. 2519-2529.

[5] Jiri Fridrich, “Image Watermarking for Tamper Detection,” in Image Processing, IEEE, vol. 2, pp. 404-408, Oct. 1998.

[6] Jiri Fridrich, “Method for Tamper Detection in Digital Images,” Multimedia and Security, Workshop at ACM Multimedia, vol. 99, Oct. 1999.

[7] Jiri Fridrich, “A hybrid watermark for tamper detection in digital images,” Signal Processing and Its Application, vol. 1, pp. 301-304, Aug. 1999.

[8] Ahmet Emir Dirik and Nasir Memon, “Image Tamper Detection based on demosaicing artifacts,” in Proceedings of IEEE International Conference on Image Processing, pp. 1481-1484, 2009.

[9] Jiaji Wu, Anand Paul, Yan Xing, Yong Fang, Jechang Jeong, Licheng Xiao and Guangming Shi, “Morphological dilation image coding with context weights prediction,” Signal Processing: Image Communication, vol. 25, issue 10, pp. 729-744, Nov. 2010.

[10] Robert M. Haralick, Stanley R. Sternberg and Xinhua Zhuang, “Image Analysis Using Mathematical Morphology,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no. 4, July. 1987.

[11] Criminisi, Antonio, Patrick Perez, and Kentaro Toyama, “Region filling and object removal by exemplar-based image inpainting,” in Image Processing, IEEE, vol. 13, no. 9, pp. 1200-1212, Sep. 2014.

[12] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles and Coloma Ballester, "Image inpainting," Proceedings of the 27th annual conference on Computer graphics and
interactive techniques, ACM Press/Addison-Wesley Publishing Co., pp. 417-424, 2000.

[13] Tony F. Chan and Jianhong Shen, “Nontexture Inpainting by Curvature-Driven Diffusions,” Journal of Visual Communication and Image Representation, vol. 12, no. 4, 2001, pp. 436-439.

[14] M. Elad, J.-L Starck, P. Querre and D.L. Donoho, “Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA),” *IEEE Transaction on Image Processing*, vol. 12, no. 8, 2003, pp. 882-889.