DIGITAL WATERMARKING FOR INTEGRITY PROTECTION OF SYNTHETIC IMAGES

Huajian Liu, Martin Steinebach, Lucilla Croce Ferri

Fraunhofer IPSI - Integrated Publication and Information Systems, Darmstadt, 64293, Germany
Email: {liu, steinebach, ferri}@ipsi.fraunhofer.de

ABSTRACT

Compared with natural images, synthetic images have less textures and more smooth areas. Therefore it is more difficult to hide information invisibly in synthetic images than in natural ones. This paper proposes a novel watermarking method to verify the integrity of synthetic images with low number of colors in palette image formats (indexed color image formats, e.g. GIF). Watermark information is embedded into randomly shuffled blocks to identify every pixel of the image. The automatic recovery of tampered areas is achieved by a quantization method. Experimental results demonstrate the ability of the proposed scheme to locate and recover tampered areas in the watermarked images.

1. INTRODUCTION

With the rapid development of powerful image processing software, it becomes very easy to manipulate a digital image. Deleting, adding or replacing of objects is possible without causing noticeable traces. As a result, no digital images can be considered trustworthy without integrity authentication. Digital watermarking is a promising solution for integrity protection of digital media content.

Some image watermarking algorithms have been proposed to authenticate image integrity. Usually, authentication patterns or content dependent features are embedded into the image as watermark information to identify the content and detect alteration. Watermark bits are commonly embedded into the least significant bits (LSB) of pixels [1-3] or transform coefficients [4]. But most of these algorithms mainly focus on natural images in gray scale or true color formats [1-4] and are not suitable for synthetic images in palette formats.

Due to the few textures and large smooth areas of synthetic images, it is more difficult to hide information invisibly. For integrity protection, however, high watermark capacity is required because every part of the image needs to be identified, and even more payload is needed if the recovery of tampered areas should be possible. Furthermore, synthetic images are usually stored in palette formats. In a palette-based image, any slight change of brightness or color in LSB of pixels or transform coefficients will cause visible artifacts and introduce new colors in the palette. Wu [5] proposed a watermarking method for binary image authentication that could be used to detect whether binary documents are manipulated or not. In that scheme, however, no location information of alteration is provided and the altered area can not be recovered.

In this work, we propose a novel watermarking scheme for integrity protection of synthetic images. Every watermark bit is utilized to identify all the pixels in one randomly shuffled block. Thus, all pixels of the image instead of blocks are identified by much less watermark bits. If the extracted watermark bit doesn't match the original one, the corresponding block is marked unverified and all pixels in this block are marked unverified as well. By the verification process, the altered area not only can be located by the unverified pixels but also can be recovered to its two-color counterpart.

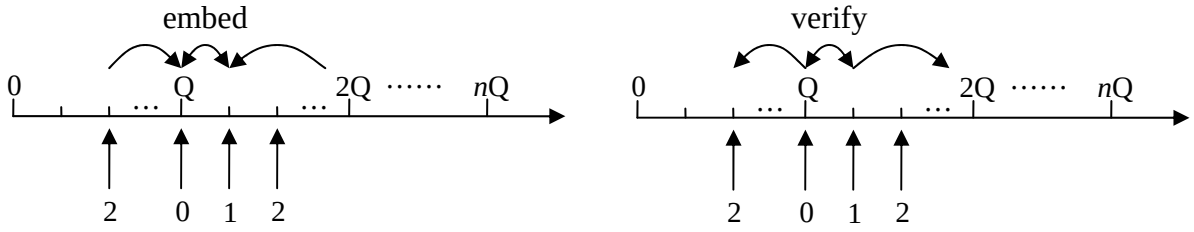
The paper is organized as follows. In Section 2, the proposed watermarking scheme for integrity protection is introduced, including both the watermark embedding and retrieval algorithms. Experimental results are given in Section 3. We conclude the paper in Section 4.

2. PROPOSED SCHEME

2.1. Watermark embedding

As mentioned in Section 1, due to the simplicity of a synthetic image, most pixels in such an image can not be changed; otherwise visible artifacts can easily be introduced. Therefore, before embedding the watermark information into a synthetic image, it should be determined first which pixels can be changed causing least noticeable artifacts.

To simplify the embedding and verifying processes, all pixels are classified into two kinds of colors, $c1$ and $c2$. According to the cover image's property, these two colors can be determined either by pixels' luminance or by their hues or by both. As described in [5], according to the local property of a 3x3 window, e.g. smoothness and connectivity, every pixel is given a score of how noticeable such a change will be.



(a) (b)

Fig. 1 (a) watermark embedding (b)image verification

In order to make the flippable pixels distribute evenly [5] and use every watermark bit to identify a set of pixels in one block, all pixels of the image are randomly shuffled before the watermark embedding. The shuffling process is controlled by a secret key K as follows.

$$I_o \xrightarrow{shuffle} I_s,$$

$$(k,l) = S((i,j),K), \tag{1}$$

where (i, j) is the pixel coordinate in the original image I_o and (k,l) the coordinate in the shuffled image I_s.

To guarantee the random distribution of all pixels and increase the accuracy of verification, the distance of adjacent pixels must be larger than a minimum *D*. The shuffled image is divided into $bSize \times bSize$ blocks and every block is used to embed one watermark bit. Small block size will increase the accuracy of a tampered area's localization and recovery, while requiring more watermark capacity and causing more changes of pixels. The block size depends on the image dimension and the number of flippable pixels. Theoretically, for a cover image of size $Height \times Width$, the smallest block size is limited to

$$bSize_{\min} = sqrt(\frac{Height \times Width \times Q}{2 \times N_{flip}}), \tag{2}$$

where N_{flip} is the number of flippable pixels in the cover image and Q is the quantization step used to embed the watermark.

The watermark information $w(n)$, used as authentication code, is generated under the control of the secret key *K*, $w(n) = G(n,K), w(n) \in \{0,1\}$. For every block, the embedding algorithm is described as in the following equations.

$$M = \frac{1}{C}\sum_{(k,l)\in block} I_s(k,l), \quad if \quad I_s(k,l) = C, \tag{3}$$

where M is the number of pixels with color C in the block and $C \in \{c1, c2\}$.

Then M is quantized by a step Q.

$$M = \left\lfloor \frac{M}{Q} \right\rfloor \cdot Q + \Delta \tag{4}$$

In order to identify two kinds of different colors when verifying the image, Q must be larger than 2. Larger Q will increase the accuracy of verification and recovery when the number of the tampered pixels in one block is larger than one. But the embedding process will introduce more changes of pixels and may decrease the watermarked image's fidelity. In our implementation, we let $Q = 3$.

As shown in Fig.1(a), if $\Delta \neq w(n)$, the pixels in the block with smaller flipping scores than a threshold T are flipped to satisfy the following expressions.

$$\Delta^* = w(n) \tag{5}$$

$$M^* = \left\lfloor \frac{M^*}{Q} \right\rfloor \cdot Q + \Delta^* \tag{6}$$

After embedding the watermark, the image is then inversely shuffled to obtain the watermarked one.

2.2. Watermark retrieval and verification

The watermarked image I_o^* is first shuffled again under the control of the proper secret key. Then the shuffled image I_s^* is divided into blocks. The watermark information $w(n)$ is also generated by the secret key *K*. Every watermark bit is retrieved as follows.

$$M^* = \frac{1}{C}\sum_{(k,l)\in block} I_s^*(k,l), \quad if \quad I_s^*(k,l) = C, \tag{7}$$

$$w^*(n) = \Delta^* = M^* - \left\lfloor \frac{M^*}{Q} \right\rfloor \cdot Q, \tag{8}$$

where $w^*(n)$ is the extracted watermark information and $C \in \{c1, c2\}$.

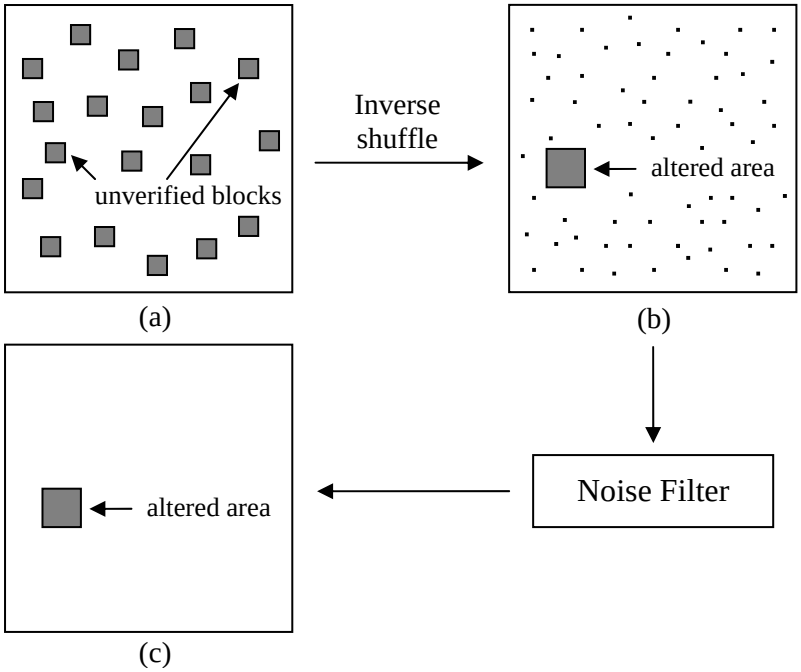


Fig. 2 (a) unverified blocks in the shuffled image (b) noise dots and altered area (c) altered area after noise filter

After obtaining the extracted watermark, the verification could be made by comparing it with the original watermark sequence as shown in Fig.1(b). For every block, if the extracted watermark bit is different from the original one, the block is marked unverified, as shown in Fig.2 (a). All of the pixels in the unverified blocks are marked as unverified pixels.

All the pixels are then mapped back to the original image and the unverified pixels will be randomly distributed over the whole image, as shown in Fig.2 (b). Only the altered area has a high density of unverified pixels. All of the isolated unverified pixels will be considered as noise dots. By a noise filter, e.g. a median filter, the altered area will be easily picked out, as shown in Fig.2 (c). A properly designed noise filter can not only filter out the noise pixels, but also can compensate for an insufficient shuffle. If the number of the altered pixels in one block is larger than one, the filter can be used to smooth out the wrong points.

The original pixels in the altered area are recovered as follows. In the case that Q is equal to 3 and M is the number of the pixels with color $c1$ in one block, the original pixels' colors are recovered as the following.

$$\begin{array}{ccc} w(n) & w^*(n) & I_o(i,j) \\ 0 & 1 & c2 \\ 0 & 2 & c1 \\ 1 & 2 & c2 \\ 1 & 0 & c1 \end{array}, \qquad (9)$$

where $w(n)$ and $w^*(n)$ are the original and extracted watermark bit. $I_o(i,j)$ is the original pixel's color.

3. EXPERIMENTAL RESULTS

We evaluate the proposed watermarking scheme by testing it on digital maps in palette image formats from the Media@Komm[1] project. In our experiments, we let $Q = 3$, $T = 2$ and a median filter with size 3×3 is used as the noise filter during verification.

For illustration, Fig.3 (a) and (b) show the original digital map (1202×876) and the watermarked version. Under normal viewing conditions, no visible artifacts can be noticed in the watermarked map. Fig.3 (c) shows a magnified part of the watermarked image. Some manipulations are then made on the watermarked map. As shown in Fig.3 (d), one curve is deleted from the watermarked map while another forged curve is added there. After the watermark retrieval and image verification using the proper secret key, the deleted curve is recovered in red color and the added curve is also marked out in blue color as shown in Fig.3 (e).

4. CONCLUSION

In this paper, we propose a novel watermarking method for integrity protection of synthetic images, which can

[1] http://www.mediakomm.net/.

locate and recover the manipulations applied to the watermarked image. In order to use less watermark bits to identify all pixels, every randomly shuffled image block is utilized to embed one watermark bit to identify all of the pixels in the block. In the verifying process, the randomly distributed unverified pixels can locate the tampered area and recover its two-color counterpart. The experimental results demonstrate the effectiveness of the proposed scheme.

5. REFERENCES

[1] P.W. Wong, "A Public Key Watermark for Image Verification and Authentication", *Proceedings of IEEE International Conference on Image Processing, Chicago*, USA, pp. 425-429, October, 1998.

[2] J. Fridrich, M. Goljan, A.C. Baldoza, "New Fragile Authentication Watermark for Images", *ICIP'2000*, Vancouver, Canada, September, 2000.

[3] J. Fridrich, "Security of Fragile Authentication Watermarking with Localization", *Proceedings of SPIE Security and Watermarking of Multimedia Contents IV*, San Jose, California, vol. 4675, January, 2002.

[4] D.A. Winne, et. al. "Digital Watermarking in Wavelet Domain with Predistortion for Authenticity Verification and Localization", *Proceedings of SPIE Security and Watermarking of Multimedia Contents IV*, San Jose, California, vol. 4675, January, 2002.

[5] M. Wu, E. Tang, B. Liu, "Data Hiding in Digital Binary Image", *IEEE International Conference on Multimedia & Expo (ICME'00)*, New York City, 2000.

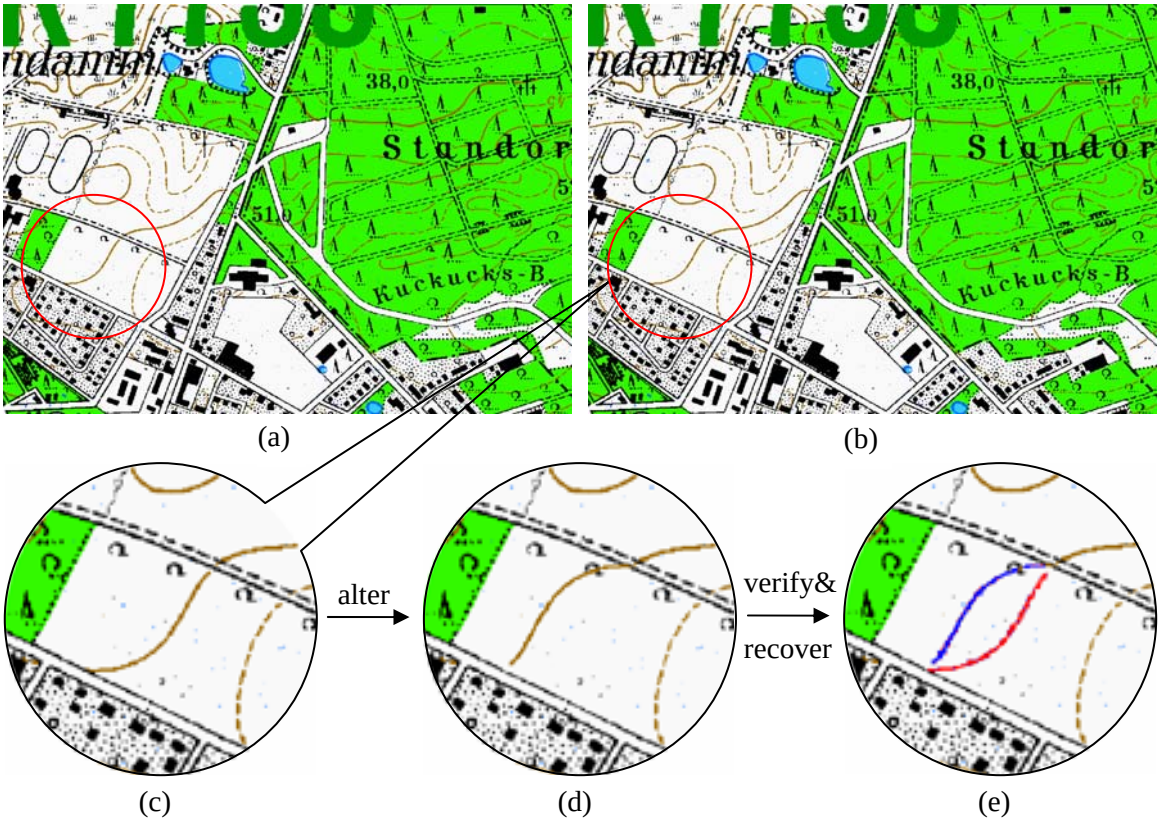


Fig. 3 (a) original map (b) watermarked map (c) magnified part from watermarked map (d) altered version (e) verified and recovered result