A Texture-based Tamper Detection Scheme by Fragile Watermark

Yazhou Liu , Wen Gao , Hongxun Yao , Shaohui Liu
Computer Science and Technology Department of Harbin Institute of Technology,
150001, Harbin, the People's Republic of China
yzliu@vilab.hit.edu.cn
wgao@ict.ac.cn
yhx@vilab.hit.edu.cn
shaohl@vilab.hit.edu.cn

Abstract

Abstract. A texture-based tamper detection scheme by fragile watermarking technique is proposed in this paper. Comparing with other fragile watermarking schemes, the highlight of our scheme is that it's rather sensitive to malicious tamper such as replacing one's face in the image by another's and at the same time it's insensitive to other legal processing such as lossy JPEG compression and brightness/contrast changes. So it is more suitable for tamper detection in practical use.

1. Introduction

With some powerful image processing softwares such as Adobe PhotoShop one can remove/replace some features in a picture easily without any detectable trace. We regard these kinds of operations as tamper. But in some cases, the images are not allowed to be done such operations, such as images for military, medical, and judicative use. The validity of the image is of most importance in these conditions. So some effective ways are need to guarantee integrity of the image. The most common means to defeat tampers is to embed a fragile watermark into the image to identify if an image has been tampered and supply localization information as to where the image has been tampered.

To illustrate some important features that tamper-detection fragile watermarks should have, we start from watermark detection scenario. Suppose an image is tested by a fragile watermark detector to identify it's integrity, there should be three kinds of response. First, if the image has been marked but not been tampered, there should be no response at all, we denote this response N; second, if the image has been marked and tampered, the output image should indicate where the alteration is, we denote this response L; the last case, if the image has not been embedded fragile watermark at all, the output image should indicate that the whole image has been changed, we denote this A. We describe above in figure 1.

Now let's guess what a forger would do. Suppose he has just finished his perfect work and left no traces at all. What will he does next? If the forger is only an amateur, he will be rather satisfied with his work and do nothing at all next. But most of fragile watermark schemes can detect this kind of tamper. So unfortunately, his work does not so perfect as he thought. Another case, if the forger is an expert in this field, he may guess that there is a fragile watermark has been embedded in the image before his work. So in order disguise his tamper, he can either adjust his alterations to make them undetectable by the watermark detector (in anther word, to make output image's response from L to N, we denote this **disguise one**) or make a unperceptual change to the other parts of the image to break up the fragile watermark in the whole image (to change the output image's response from L to A, we denote this **disguise two**). So that the mark detector can not tell if the image has been tampered or the image has not been embed a watermark at all (so in this case, the forger may argue that the original image has not been marked and there is no tamper at all). We describe above in figure 2. For in most cases, the forger does not know what the fragile watermark embed scheme has been used or what the embedding secret key is, so it's so difficult for him to come up with an almost transparent alteration by the first means. But he can easily achieve his goal by the second means. For instance, if the fragile watermark based on LSB technique, what a forger should do is just disturb all the LSBs or any operation with low-pass character. So from this point of view, we can easily make a conclusion that a tamper-detection watermark scheme is a effective one only when it's ROBUST enough to the disguise two.

In section 2, we will analyze some previous fragile watermark schemes and their capability in tamper detection. We will give details of our temper detection scheme in section 3 and provide our experimental result in section 4.

2. Previous works

The prime goal of a fragile watermark is to detect changes in a image, but we should note that not all changes in a image can be called Tamper. For instance, if a image used to be compressed by JPEG, it's changed but not tampered; if someone has increased it's brightness slightly, it's changed but not tampered; but if someone replaced a person's face with another's, the image has been tampered. Tamper is the apparent changes in relative small scales. So not all fragile watermark schemes are suitable for tamper detection.

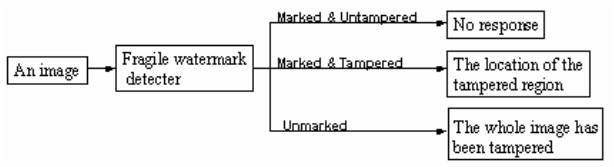

Fig. 1 Three kinds of outputs when an image is tested by a fragile watermark detector

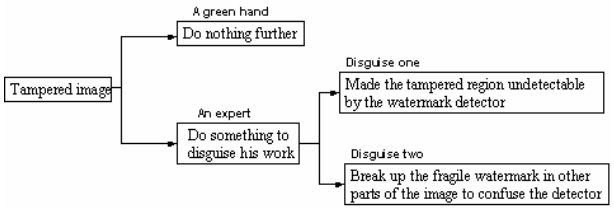

Fig.2 Two kinds of disguise may be used by he forger

Some fragile watermark schemes based on least significant bit (LSB) technique can detect changes in images very effectively. In[2], Wong proposed a public key fragile watermark scheme which divide the image into blocks and then embed signature for each block in the LSBs. This scheme is so sensitive that can detect any change made to an image including changes in pixel values. Mehmet in[3] extended Wong's work by forming a multi-level hierarchical block structure which improved scheme's robustness and localization. Because these schemes based on LSB, they are so sensitive to changes that they are not suitable for tamper detection. Just as mentioned above, by disturbing the LSBs in whole image or any operation with low-pass character, the forger can break up the fragile watermark of the image. So in another word, these schemes are susceptible to disguise two. Other fragile watermark schemes test changes by hash function. These schemes using the hashed digest of the original signal to decide the authenticity of the content. Such as row-column hash function (RCHF) technique and block-base hash function (BBHF) technique in[8]. But their disadvantage is just as the LSB's and not suitable for tamper detection.

A DCT based means was proposed by Fridrich[4] [5]. They divide image into 64x64 blocks and embed the watermark in each block's DCT domain by spread spectrum technique. This scheme is robust to the second kind disguise but not very capable of tamper detection because it can not detect smaller tamper effectively. It sacrifice too much localization accuracy to improve robustness.

3. Proposed scheme

3.1 Overview of texture based block-linking scheme

From previous analysis, we can see that a tamper-detection fragile watermark must be not only fragile in local scale to identify tamper but also robust in global scale to defeat disguise two. So it's important for us to find a balance point between fragile and robust. The goal of tamper is to change some meaningful features in an image and the content of the image is represented by the texture. So we choose texture as criteria of tamper. In our texture based block-linking (TBBL)scheme, we divide the image into 8x8 blocks and classify them into four groups according to their texture character and then liking them together by modulation of their low/middle DCT coefficients.

3.2 Fragile watermark embedding procedure

Classify the image We divide the image into 8x8 blocks and then classify all blocks into four groups according to their texture character. And the four groups are: type zero, no texture group which means that the variances of the blocks in this group are relatively small; type one, vertical texture group for we can see apparent vertical texture in the blocks of this group; type two is horizontal texture group; type three include all the blocks that don't belong to the other three. We chose texture as our cluster criteria because most legal alterations in global scale such as adjust the brightness/contrast in a image can not change the texture characters of blocks so that these alteration can not affect the classification. But at the same time, this classification is rather sensitive to tamper such as removing/adding some features from/to an image because they all changed the texture of the blocks. One may argue that if he can replace a person's face with another's which almost has the same size, posture and distribution as the original one, the texture of the blocks occupied by the face may not be changed. This case is possible, but it will probably affect the blocks' pointer which will be embedded in the next step.

Embed pointer In this step we will embed a pointer into each block to link all the blocks together. The pointer of a block should point to it's preceding block, in another word, the pointer's value should be equal to the block's type which is in front of it. We embedded the pointer into the DCT domain by modulating their coefficients of middle/low frequency (exclude DC). Because there are four types of blocks, we only need two bits to denote a block's type number, which means that we only need to change two coefficients. In our scheme we select six coefficients to increase robustness. Just as direct sequence spread spectrum techniques, we use three bits to denote one bit. Denoting the i-th block B_i, we generate a seed number S_i by hash function $H(\cdot)$,

$$S_i = H(k, n, m)$$

where k is the secret key, n the B_i's column number and m the B_i's row number. By S_i we select six coefficients randomly in middle/low frequency, we denote they are $c_1,c_2,\cdots,c_6$, and quantize them by standard JPEG quantization table. Then divide them into two triples and modify these two triples to the same parity respectively to denote two bits by adding or subtracting half of their corresponding quantization step size. For instance, we denote even/odd number to 0/1 and classify the six coefficients as c_1,c_2,c_3, and c_4 c_5,c_6. If the previous block's type is two, what should we do is just to modify the quantization indexes of the first triple to their nearest odd numbers and the second triple's to their nearest even numbers. Then the six coefficients were put back their corresponding locations and the image was transform to the spatial domain. Since the JPEG exploit the masking phenomena in human visual system(HVS), previous modification will not leave perceptually noticeable trace in the image and will not affect the blocks' clustering. Because our pointers are embedded in the middle/low frequency coefficients and our spread-spectrum like embedding scheme so they are robust to most kinds of common operations such as JPEG, brightness/contrast changes. And at the same time, these points are rather fragile to tamper. For instance, if we select our pointers from 35 middle/low frequency coefficients, the probability of that a forged block has the same pointer as the original one is less than $3.1\text{x}10^{-8}$, and if the forger attempt to break up all pointers in the image he will have to disturb all the coefficients which will cause serious degradation of image quality.

3.3 Fragile watermark extraction procedure

In this procedure, we divide the image into 8x8 blocks firstly, and then we can get each block's type and pointer just as mentioned in previous section. We check the whole image block by block along the link. If one block has been tampered, we denote it to B_i, it's pointer or type (maybe both) must has been changed. So if B_i's pointer can not point to it's previous block B_{i-1} and at the same time it's type does not identify to it's following block B_{i+1}'s pointer, block B_i must has been tampered. If B_i's pointer can not point to B_{i-1} but B_{i+1}'s pointer can point to B_i or reversely, we may consider block B_i at edge of the tampered region. By this means we can check out all the tampered blocks.

3.4 Analysis of robustness to legal operations

From previous instruction of our scheme, we can see that it's robustness to modest legal operations based on two key factors: the classification's robustness and the pointers' robustness. Just as has been analyzed in preceding sections, the classification is robust because most modest legal operations can not change a block's texture. In this section, we will focus on proofing the robustness of the pointers.

From the pointers' embedding procedure, we can see that they are robust to JPEG compression. Now we will proof their robustness to brightness/contrast adjustment. If we denote the original image I, the changed image I' and the brightness/contrast modification ΔE. We can model these modification as following:

$$I' = I + \Delta E \qquad (1)$$

In brightness modification, we can regard ΔE as a image with all it's pixels' value equal, so it's all DCT coefficients are zero except DC and can not affect out pointers in DCT domain at all. We can deduce the equation (1) as following:

$$\begin{aligned} DCT(I') &= DCT(I + \Delta E) \\ &= DCT(I) + DCT(\Delta E) \\ &= DCT(I) \end{aligned}$$

As for modest contrast modification, in a relative small scale (such as our 8x8 block), the coefficients of $DCT(\Delta E)$ are relative small comparing with $DCT(I)$'s, and the pointers will not be affected greatly.

4. Experimental result

In order to evaluate the proposed fragile watermarking scheme, we embed the watermark into the image of "F16" (256x256), which yield the watermarked image seen **Fig. 3**. Watermarked image is then tampered by removing the logo " U.S. AIR FORCE " to yield the image seen in **Fig. 4**. Then the tampered image is tested by our fragile watermark detector, the tampered region can be can be located by the output image seen in **Fig. 5**. Next we adjust the tampered image's brightness and contrast by –40% respectively by Photoshop 6.0, and then the manipulated region can still be find out precisely, whose output image seen **Fig. 6** and **Fig. 7**.

5. Reference

[1] Ingemar J. Cox, Joe Kilian, F. Thomson Leighton, and Talal Shamoon "*Secure Spread Spectrum Watermarking for Multimedia"* in IEEE Trans. on Image Processing, 6, 12, 1673-1687, 1997.

[2] P.W. Wong, "*A public key watermark for image verification and authentication*" in Proceedings of IEEE International Conference on Image Processing, Chicago, USA, October 4-7, 1998, pp. 425–429.

[3] Mehmet U. Celik , Gaurav Sharma , Eli Saber , and A. Murat Tekalp, "*A hierarchical image authentication watermark with improved localization and security*," in

Proceedings of IEEE International Conference on Image Processing (ICIP-2001), Thessaloniki, Greece, October 7-10, 2001, vol.2, pp. 502 –505.

[4] J. Fridrich, "*Methods for Detecting Changes in Digital Images*", Proc. of The 6th IEEE International Workshop on Intelligent Signal Processing and Communication Systems (ISPACS'98), Melbourne, Australia, 4–6 November 1998.

[5] J. Fridrich, "*Image watermarking for tamper detection*" in Proceedings of IEEE International Conference on Image Processing (ICIP 98), Chicago, USA , Volume: 2 , October 4-7 ,1998 , vol.2, pp. 404 –408.

[6] R. B. Wolfgang, and E. J. Delp, "Fragile Watermarking Using the VW2D Watermark," Proceeding of the IS&T/SPIE Conference on Security and Watermarking of Multimedia Contents, pp.204-213, San Jose, CA, January 1999.

Fig. 3 Original watermarked image of "F16", 256 x 256

Fig.4 The tampered image, "U.S. AIR FORCE" has been removed

Fig. 5 The output image of the fragile watermark detector

Fig.6 The output image after the tampered image's brightness was adjusted by –40% (Photoshop 6.0)

Fig.7 The output image after the tampered image's contrast was adjusted by –40% (Photoshop 6.0)