International Journal of Electrical, Electronics and Computer Engineering **1**(2): 7-11(2012)

ISSN No. (Online) : 2277-2626

A Survey of Digital Water Marking Embedded in Colour Image and Security Attack

Yash Kshirsagar, Anshul Awasthi and Mamta Sood

Department of Electronics and Communication TIT, Bhopal, (MP)

(Recieved 10 May, 2012 Accepted 25 June, 2012)

ABSTRACT : The rapid growth of the Internet in the past years has rapidly increased the availability of digital data such as audio, images and videos to the public. As we have witnessed in the past few months, the problem of protecting multimedia information becomes more and more important and a lot of copyright owners are concerned about protecting any illegal duplication of their data or work. Some serious work needs to be done in order to maintain the availability of multimedia information. In this paper we have survey of different approach of digital water marking technique such as DCT based technique, pixel Interchanged based technique, rotation of pixel technique and amplitude of the quaternion technique and related technique of digital water marking for the data truncation.

Keywords: Digital water marking, DCT, pixel rotation.

I. INTRODUCTION

A watermark is a recognizable image or pattern in paper that appears as various shades of lightness/darkness when viewed by transmitted light (or when viewed by reflected light, atop a dark background), caused by thickness variations in the paper.A watermark stored in a data file refers to a method for ensuring data integrity which combines aspects of data hashing and digital watermarking. Both are useful for tamper detection, though each has its own advantages and disadvantages. Digital watermarking is the process of possibly irreversibly embedding information into a digital signal. The signal may be audio, pictures or video, for example. If the signal is copied, then the information is also carried in the copy. Traditional watermarking schemes accomplish watermark embedding into the host signals by sacrificing the imperceptible host information. For instance, one can replace the least significant bit (LSB) plane of the host signal with a fragile watermark. Though easily realized, such strategies are not suitable for some special application scenarios, such as the medical fields, remote sensing, and military imagery, where data integrity verification or authentication is only allowed under the condition that no host information is distorted. As a solution to this problem, researchers proposed the concept of reversible watermarking algorithms. It means that these algorithms not only can realize copyright protection and content certification by embedding the assigned watermarks into host signals, but also can completely recover the original host signals by extraction algorithm. This concept was first proposed in a patent of America by Honsinger [1] in 1999. Jessica [2] proposed the modulo addition method in space domain, but this algorithm had a defect of pixel value overflow. Guorong Xuan and Y.Q. Shi [3] made reversible data compression in reversible integer wavelet domain, and embedded the watermark in vacated bit space. Michiel van der Veen [4] proposed an algorithm based on bit shift in information security meeting of SPIE in 2003. Ni and Shi [5] proposed a new reversible watermarking method by changing the pixel value histogram in space domain so far, reversible watermarking schemes can be classified into three types: by difference expansion, by data compression, and by histogram bin exchanging [6]. The embedding capacity, robustness, imperceptibility and calculating complexity are the basic criterions of the reversible watermarking algorithms. The schemes using difference expansion are weak in robustness because the destroyed location map may cause mismatching. Any loss of the compressed data may destroy the whole embedded data because the most data compression techniques cannot resist any distortions. The algorithms by histogram bin exchanging may have higher robustness, but have low embedding capacity at the same time. In recent years, a concept named near reversible watermarking was discussed widely. It means that data modifications can be accepted, supposing that the value of pixels difference between recovered and original host data are within a maximum user-defined distance. If this bound is sufficiently low, the watermarking embedding and extraction process can be considered as near reversible. M. Barni and F. Bartolini [7] proposed a near-reversible digital watermarking for copyright protection of remote sensing images. By forcing a maximum absolute difference between the original and watermarked scene, the near-reversible paradigm made it possible to decrease the effect of watermarking on remote sensing applications to be carried out on the images. Farid Ahmed and Ira S. Moskowitz [8] proposed a frequency domain digital watermarking technique that was a semi-reversible watermark for medical image Authentication. This paper proposes in V section. Section I gives a introduction of digital watermarking.

Section II gives a near reversible image water marking. Section III gives a Quaternion frequency algorithm. Section IV gives geometric attack in watermarking. Section V gives a conclusion

II. NEAR REVERSIBLE IMAGE WATER MARKING

By Zha, Jiang [1] gives the near reversible watermarking embedding and extraction algorithm.

Step 1 : Read into the image files, get the data matrixes of host and watermark image $I(i, j)$ and watermark image $W(i, j)$, write the size of watermark into the head of the host image. $I(i, j)$, $i = 1 : M$, $j = 1 : N$, M and N are the row and column of host image respectively. $W(i, j)$, $i = 1 : m$, $j = 1 : n$, m and n are the row and column of watermark respectively.

Step 2 : Calculate the ratio R of host image and watermark image by formula (1).

$$R = (M \times N)\ (m \times n) \qquad \text{... (1)}$$

Step 3 : Divide the host image into small blocks according to the value of R.

1. If $R < 8$, the size of watermark image extends the capacity of host image, and the algorithm stop;
2. If $R \geq 8$, the watermark image can be embedded into the host image. The larger R is, the less the host image data will be lost.

Step 4 : Embed the watermark data into the divided small blocks. For example, when $R \geq 64$, the host image is divided into a series of 8 × 8 blocks, and one byte of watermark data can be embedded into a small block. The embedding procedures are as follows :

The 8th row secondary LSB data of the host image block replace the LSB data of the same row, and the 8th row original LSB data is discarded. Then, the 7th row secondary LSB data replace the 8th row secondary LSB data, the 6th row secondary LSB data replace the 7th row secondary LSB data…, the 1st row secondary LSB data replace the 2nd row secondary LSB data, and one byte of watermark data replace the 1st row secondary LSB data of the host image block. If one byte of watermark data are :

$W(8)$ = [1 1 1 1 1 1 1 1]. The secondary LSB data of original host image block are :

$$I(8,8) = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The secondary LSB data of the watermarked image block are :

$$I'(8,8) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The 8th row LSB data of the watermarked image block are:

$P(8)$ = [1 1 0 1 1 0 0 1] and the 8th row LSB data of original is lost.

Step 5 : end

Extraction procedure. The extraction algorithm is a reverse process to the embedding process. The detailed extraction procedures are as follows :

Step 1 : Extract the size of the watermark from the head of host image data.

Step 2 : Calculate the value of R by (1) and divide the watermarked image to small blocks according to the value of R.

Step 3 : Extract the 1st row secondary LSB data of every blocks and recompose the watermark image.

Step 4 : Recover the original host image. The detailed procedures are as follows : The 2nd row secondary LSB data replaces the 1st row secondary LSB data, the 3rd row secondary LSB data replaces the 2nd row secondary LSB data, the 4th row secondary LSB data replaces the 3rd row secondary LSB data…, the 8th row secondary LSB data replaces the 7th row secondary LSB.

Data and the 8th row LSB data replaces secondary LSB data of the same row.

Step 5 : end.

Section III Quaternion frequency algorithm. By Sun Jing, Yang Jing-yu1 [2] proposed a method Quaternion frequency algorithm Quaternion, also called hyper-complex numbers, is an extension concept of complex sets, put forward by British mathematician Hamilton as early and *E*ll since 1990s last century. A quaternion also has four components :

$$q = a_0 + a_1 i + a_2 j + a_3 k \qquad \text{... (1)}$$

where 0 1 2 a, a, a and 3 a are real numbers, i, j and k are imaginary operators presenting the following properties :

$$\begin{cases} t^2 = j^2 = k^2 = -1 \\ ij = -ji = k,\ jk = -ki = i,\ ki = -ik = j \end{cases} \qquad \text{... (2)}$$

A quaternion has a real part 0 $S(q) = a$ and an imaginary part. The latter has three components and thus, can be used as a vector quantity, often denoted by $V\ q\ a\ i\ a\ j\ a\ k$ 1 2 3( ) = + +.

The whole quaternion may be represented by the sum of its scalar and vector parts as $q = S(q) + V(q)$. A quaternion with a zero real or scalar part is called a pure quaternion Euler's formula for the complex exponential generalizes to hypercomplex form $q = |m|$ (cosΦ + μsinΦ), where

$\|m| = a + a + a + a$ is called the modulus of the quaternion. μ and Φ are referred to as the eigenaxis eigenangle direction in 3-space of the vector part and may be regarded as a true generalization of the complex operator i, since μ2 = –1. Φ is analogous to the argument of a complex number, but is unique only in the range [0, π] [4]. Using this representation, a color image $f(x, y)$, sized $M \times N$ could be considered as an array of pure quaternion numbers (*i.e.* with no real parts) :

$$f^{q}(x, y) = R(x, y)i + G(x, y)j + B(x, y)k \qquad \text{... (3)}$$

where $R(x, y)$, $G(x, y)$ and $B(x, y)$ represent classical Red, Green and Blue color components.

In [5] Sangwine and *E*ll have defined the left form of *QFT* as :

$$QFT^{q}_{f\ L}(u,v) = S\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} e^{-\mu\left(\frac{xv}{M}+\frac{yu}{N}\right)} f^{q}(x,y) \qquad \text{... (4)}$$

III. BLOCK SELECTION BASED ON VISUAL CAPACITY

Human visual system (HVS) can't perceive the embedded watermarks' existence when the embedding strength is lower than a certain threshold which differs from the regions of the image. Generally, there exists the affection of visual mask that the higher threshold is accompanied with greatly varied background and highly textured information. In order to pick out the potential blocks which have lager visual capacity to embed the watermarks adaptively, we calculate the ununiformity (represented by d) of each unit small block by the formula shown as :

$$d(Q_z) = \frac{1}{n^2}.\sum_{(i,j)\in Q_z} \frac{abs[f(i,j)-m_z]}{abs(m_k^{1+a})} \qquad \text{... (5)}$$

For calculating convenience, the host image matrix is supposed to be a square matrix, sized $M \times M$, where $n \times n$ is the size of each unit small block. Q_k represents a certain unit small block with mean value km and α is the revised weighted factor with value 0.6 to 0.7 and abs represents calculation of the modulus. The higher of d, the lager visual capacity is, means the worse uniformity of the block and the more complex of its texture.

Section IV Geometric attack in watermarking. By Hang SU, Chuqing [3] LV proposed a new technique for geometrical attack for digital water marking embedding process YUV convert Our bill images scanned in may not be gray-scale images, and our experiment shows that DCT on the *Y*-components in YUV has better performance than modifying whole color components of RGB. When facing GRB images,we convert it to YUV domain and use the *Y*-components (luminance component) to hide watermark information. After conversion we save the luminance components array of RGB images. For gray-scale images, we save its gray-scale array. Block DCT.

1. Divide blocks Divide effective information array extracted in prior step into K32 × 32 sub blocks $K = (m/32) \times (n/32)$, and save in arrays set $F[k, 1024]$.

2. Operate watermark. We read and save binary watermark information in array $W[256]$. For each sub blocks, we divide exert the twodimensional DCT and select 256 coefficients of the middle frequencies. Using (1).

$$S(v,u) = \frac{C(v)}{2}\frac{C(u)}{2}A$$

$$A = \sum_{x=0}^{31}\sum_{x=0}^{31} S(y,x)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

$$u, v = 0...31$$

$$C\left(\frac{u}{v}\right) = \frac{1}{\sqrt{2}}, \frac{u}{v} = 0$$

$$C\left(\frac{u}{v}\right) = 1, \frac{u}{v} > 0 \qquad \text{... (1)}$$

Embed strategy

Step 1 : We assume that p represents the row number of sub blocks and j represents column number of sub blocks, and ensure the point $F[k, p\ {*}32 + q]$ in the middle frequency of $F[K]$ and assign 0 to the variable b.

Step 2 : We count each value in $[p, q]$ points of all K sub blocks. the positive value number is recorded in variable pos and negative in neg. If $W[b]$ represents white, then check whether the result of (pos-neg) bigger than a threshold we regulated before and go to step 3.

Step 3 : If the result is bigger than the threshold, add 1 to b and back to step 2, else absolute the value of $[p, q]$ for each sub blocks one by one until the result bigger than threshold.

Step 4 : If *W*[*b*] represents black, we execute adverse Operations.

For each sub blocks we exert IDCT conversion operation and merge the n sub blocks (if we cut some data when dividing sub blocks, also merge these data in this step using (2).

$$S(y,x)=\sum_{v=0}^{31}\frac{C(v)}{2}A$$

$$A=\sum_{u=0}^{31}\frac{C(u)}{2}S(y,x)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

$$y, x = 0...31$$

$$C\left(\frac{u}{v}\right)=\frac{1}{\sqrt{2}},\frac{u}{v}=0$$

$$C\left(\frac{u}{v}\right)=1,\frac{u}{v}>0$$

If we have executed YUV convert operation before, then exert converse YUV formula to restitute RGB images. Finally, we got the watermarked image. Fig. 1 illustrates embedding process

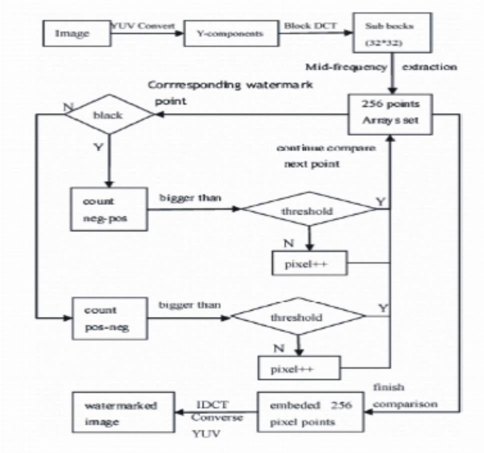

Fig. 1. Watennark embedding processing.

IV. WATERMARK EXTRACTION PROCESS

If the watermarked image to be detected has been rotated we need to redress it to help improve our detection result correct rate White border cutting To improve the efficiency and minimize the detection sphere, we first cut down the waste white border and generate a new minimum rectangular image containing the test bill image data. *B*. Oblique angle detection Assuming image's height *h* and width *w*, *p*(*x*, *y*) represent the effective value (*Y*-components for RGB and gray-scale value for gray-scale images) of point (*x*, *y*) in this image. Border extraction When detecting the upper border line and bottom border line, we select areas *A*l{(*x*, *y*)1 *w*/*lO* : *Sx* : S9 *w*ll0, O : *Sy* : *Sh*/3} and *A*2{(*x*, *y*)1 *w*/1O : *sx* : *S*9*w*/*IO*, 2*h*/3 : *Sy* : *Sh* }; detect from the top downward when dealing with upper border and form the bottom upward when dealing with bottom border. We list the upper border detection process below :

Step 1 : Initialize an arrays set *B*. From the first line to the last in area *A*l , detect the first not white pixel point and save it in the first border array *b*(*b*[*n*] represent the *n* detected point's pixel value *p*(*x*, *y*").

Step 2 : At the same time we calculate the distance of the detected point and its prior one. We set a threshold 3. If the distance of the current detected point and the prior point longer than the threshold, we assume that there may be some interference. Then initialize a new array *b* in arrays set *B* and put the current detected point in new array *b*. If the distance is shorter than the threshold, we assume that they are continuing points and but the current detected point in current array *b* (the array where we save prior detected point.).

Step 3 : When we finish detection, we got the arrays set *B* where saving many sub arrays *b*. We choose the longest array length sub-array *b*, as it possesses the most continuing points and should be the most stable array. And it is the array we use to calculate the upper border oblique angle. Equally, we detect the bottom border line in *A*2 and the left and right border line in area *A*3{(*x*, *y*)1 0 : *Sx* : *Sw*/3, *hIIO* : *Sy* : *S*9*hIIO* } and *A*4{(*x*, *y*) 12*w*/3 : *Sx* : *sw*, *hIIO* : *Sy* : *S*9*h*/*IO*}.

D. Angle judgment

Step 1 : Assume point $A(x_l, y_l)$ and $B(x_2, y_2)$ as the begin and end point of upper border line array *b*. we use *b*[*n*]*x*, *b*[*n*]*y* to represent the *x* and *y* value of the *n* point in array *b*. *L*l refers to *b*. Length. We calculate point *A* and point *B* using (3).

x_l = (*b*[5]x + *b*[6]*x* + *b*[7]*x* + *b*[8]*x* + *b*[9]*x*)/5

y_l = (*b*[5]*y* + *b*[6]*y* + *b*[7]*y* + *b*[8]*y* + *b*[9]*y*)/5

x_2 = (*b*[*L*1 – 5]*x* + *b*[*L*1 – 6]*x* + *b*[*L*1 – 7]*x* + *b*[*L*l – 8]*x* + *b*[*L*l – 9]*x*)/5

y_2 = (*b*[*L*1 – 5]*y* + *b*[*L*1 – 6]*y* + *b*[*L*1 – 7]*y* + *b*[*L*1 – 8]*y* + *b*[*L*l – 9]*y*)/5

Step 2 : Calculating oblique angle B_l using (4) tan(B_1) = (y_2 – y_l)/(x_2 – *x*l).

Similarly we calculate other three border line's oblique angle (B_2, B_3, B_4) and their length (L_2, L_3, L_4).

Step 3 : Calculate the average oblique angle of this image. Oblique angle is calculated using (5) : 8 = (B_l × L_l) + (B_2 × L_2) + (83 × *L*3) + (84 × *L*4)/(*L*l + *L*2 + *L*3 + *L*4*y* (5)

E. Image redress

Step 1 : Set canvas size. In order to not lose pixel information, we must enlarge ordinary image according to calculated rotation angle 9. Assume L represent the longest border length of ordinary image, then the new canvas' border length is *Lx* (l/cos9).

Step 2 : Calculate pixel range and inverse transformation.We calculate corresponding pixel position in new canvas using (6). *i*' = *i*' cos9 + *j*'sin9, *j'* = –*i*'sin9 + *j*'cos9 (6).

We fix the pixel position range using (7). *i*' = *i'* cos9 – *j*'sin9, *j*' = *i*'sin9 + *j*'cos9 (7).

Step 3 : Pixel refine and fix canvas point. If the value of *i* or *j* exceed the range we calculated before, then regard the pixel point as canvas point and set it to white pixel point.

F. Watermark xtraction

Step 1 : Execute same operation in Embedding process 1 and 2 for images to be detected.

Step 2 : From *b* = O to *b* = 255, check each [*p*, *q*] points in arrays set *F*'[*k*, 1024]. If the positive value number bigger than negative value number then assume watermark array *W*[*b*] represent white, otherwise black. And in this way we extract whole watermark images.

V. CONCLUSION

In this paper we study of various digital water marking technique but in all these technique we have find better three technique near reversible watermarking algorithm, geometric attack algorithm and quaternion algorithm all these algorithm are very efficient but suffering from time complexity analysis. And also facing problem such as low data embedding and loss of data. now we have design a better algorithm for digital water marking for above suffered problem.

REFERENCES

[1] Bin zhang 1, 2, yang xin1, xin-xin niu1, kai-guo yuan1, hui-bai jiang3 "a near reversible image watermarking algorithm" in *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao,* 11-14 July (2010).

[2] Sun Jing 1, 2 Yang Jing-yu1 "Quaternion Frequency Watermarking Algorithm for Color Images "in in *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics,* Qingdao, 11-14 July (2010).

[3] Hang SU, Chuqing LV, Yanbing JI, Yulin WANG "A Watermarking E-note Technique against Geometric Attacks" in 2nd *International Conference on Mechanical and Electronics Engineering* (ICMEE 2010).

[4] BU Fei-yu, LIU CHang-song, DING Xiao-qing. A Fast Algorithm of Skew Detection and Correction on Gray Business Card Image, *Journal of Chinese information processing,* Vol. **18** No. 1(2007).

[5] Basis for Digital Image Processing, Zhu Hong, Science Press, Beijing China.

[6] Xiaolin Li, Yan Feng. Image rotation policy based on DSP althorithm. *The Application of Power Electronics* TP39I.4 1, vol. **I**. (2008).

[7] Cao Wan-peng Che Ren-sheng Ye Dong Huang Qing-cheng A Fuzzy Edge Detection ethod Based on the Wavelet's Module Magnitude and Gradient Direction Proceedings of 6th *International Symposium on Test and Measurement* (Volume **6**), (2005).

[8] Michiel van der Veen *et al*. High Capacity Reversible Watermarking for Audio. In Processing of the SPIE, 5020: 1-11(2003).

[9] Z. Ni, Y.Q. Shi, N. Ansari and W Su. Reversible Data Hiding. *IEEE Processing of SCAS'03*, **2**: II-912 II-915(2003).

[10] Jen-Bang Feng, Iuon-Chang Lin *et al*. Reversible Watermarking: Current Status and Key Issues. *International Journal of Network Security*, Vol. **2**, No. 3, PP.161-171, (2006).

[11] M. Barni, F. Bartolini, V. Cappellini, E. Magli, G. Olmo. Near-reversible Digital Watermarking for Copyright Protection of Remote Sensing Images. Geoscience and Remote Sensing Symposium, IGARSS'02, (2002).

[12] Ahmed and Ira S. Moskowitz. A semi-reversible watermark for medical image uthentication. Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference Arlington, Virginia, USA, April 2-4, (2006).

[13] Jessica Fridrich, Miroslav Goljan, Rui Du. Lossless data embedding new paradigm in digital watermarking.