Data Hiding and Image Authentication for Color-Palette Images

Chih-Yang Yin (殷志揚) and Wen-Hsiang Tsai (蔡文祥)

Department of Computer & Information Science
National Chiao Tung University
1001 Ta Hsueh Rd., Hsinchu, Taiwan 300, R.O.C.
Tel: 886-3-5720631
Email:whtsai@cis.nctu.edu.tw

Abstract

In this paper, a novel method for simultaneously embedding binary data and fragile watermark signal within digital color-palette images is proposed. The color-palette of an image is first re-sorted from dark-to-bright, which makes the palette colors in the unique order even when the image is re-saved by other applications. Moreover a data hiding method for replacing the palette index is designed. The even-odd relation between two pixels is utilized for hiding any form of binary data. Finally, a novel method that embeds fragile watermark signals within color-palette images is also proposed in this study. The fragile watermark signals are embedded in the central pixel of every 3×3 block. Experimental results show the feasibility and practicability of the proposed approaches.

(KEYWORD: Data Hiding, Image Authentication, Fragile Watermark, Color-Palette Image)

1. Introduction

With the rapid growth of digital techniques, the life-style of human beings, the social structures, and the contents of civilization, have changed hugely. It is just started, and is unstoppable and unavoidable. In the transition from the old age to the digital world, the digitization of a civilization's asset has become a very important issue. Digitization does provide many advantages. First, digital data could be preserved permanently without any degradation. And data distribution via the Internet is easy and fast. The management of digital data is easy, too, which facilitates people to quickly find what they want.

But digitization, on the other hand, does reveal some new problems, too. Generally speaking, it is difficult to determine the fidelity or integrity of digital visual data because the development of digital processing technologies has made production of forgery an easy job. The problem originates from the intrinsic features of digital information: (1) making copies is easy and cheap; (2) each copy is exactly the same as the original one; (3) distribution via the Internet is easy and fast. The ease of copying and editing facilitates unauthorized use, misappropriation, and misrepresentation, which have brought un-estimated intellectual property losses. Thus, there is a great interest in developing technologies that help to protect the integrity of a digital medium and the copyright of its owner.

Many techniques have been developed for resolving the problem. Data hiding in images is one of the techniques that embed information inside an image, called the cover image in this study, to provide data security. The goal of data hiding is to embed a secret message into a cover image without making perceptual changes to the cover image under human observation. An authorized user can extract the secret message from the processed cover image, called the stego-image in this study. An unauthorized user cannot even sense the existence of the secret message in the stego-image, and message protection is hence achieved.

Many different image data hiding methods have been proposed during the last few years

[1][5][6][8][13][14] and most of them can be seen as substitution systems. Such methods try to substitute redundant parts of the image with the secret message. The main disadvantage is the relative weakness against modifications. Recently, the development of new robust watermarking techniques led to advances in the construction of robust and secure image data hiding systems.

Image authentication, on the contrary, is a technique for verifying the integrity and fidelity of an image. The uses of fragile watermarks and semi-fragile watermarks are two of the techniques developed for image authentication. A fragile watermark is a kind of watermark that is designed to be easily destroyed if the watermarked image is manipulated in the slightest manner. Image authentication can be achieved by inspecting whether the embedded signal is destroyed. But in some applications, on the other hand, "information preserving" operations (such as JPEG lossy compression) should be considered as legal operations. A semi-fragile watermark, which tells the difference between an information preserving operation and an information altering one (such as feature replacement), is hence designed to point out real tampering and ignore legal operations in the authentication process.

In the recent years, proposed systems that are used for verifying the authenticity of a digital image may be categorized into two types, the signature system [4] and the fragile watermark system. In a signature system, a digest of the data to be authenticated is obtained by the use of cryptographic hash functions. The recipient verifies the signature by examining the digest of the data and using a verification algorithm to determine if the data are authentic. A disadvantage of the signature system is that the additional signature must be stored and transmitted separately from the protected image. A fragile watermark system, on the contrary, embeds the authentication information inside the image and provides the ability to localize the altered areas within the image. Fragile watermark systems can be classified into two types, spatial-domain fragile watermarking system [2][10][11][12] or transformed-domain watermarking one [3][7][9]. In this study, we propose a novel method for simultaneously embedding binary data and fragile watermark signal within digital color-palette images.

2. Characteristics of Color-palette Images and a Palette-sorting Algorithm

In order to hide information within a color-palette image, some characteristics of them must be identified:

1. every color-palette image contains at most 256 colors;
2. a color palette is used for storing 256 colors used in the image;
3. every pixel can be thought as an index number, which is a reference to the color palette, and so every pixel needs one byte of storage.

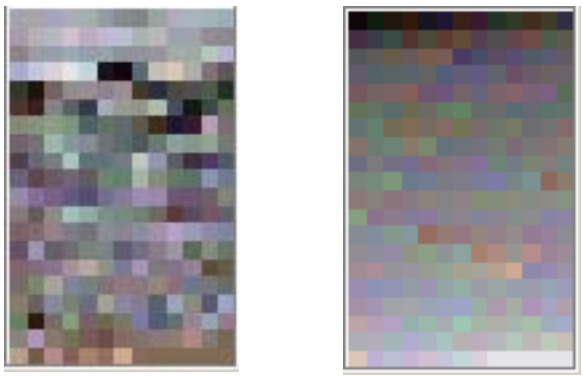
In this study, the proposed method is closely dependent on the pixel's index number within the color palette. But almost every image processing software has a different method to store the palette data. Even when we just re-save the same image by different software, the palette data may be different from the original one. As a result, we will first sort the color palette before the hiding process to make it in the dark-to-bright order. After sorting, we may reference the same palette data. This will facilitate the subsequent hiding process.

Let $p_i = (R_i, G_i, B_i)$ be the *i*-th color within the color palette, where $0 \le i \le 255$. To sort the palette colors, we first calculate

$$\mu(p_i) = \omega_1 \times R_i^2 + \omega_2 \times G_i^2 + \omega_3 \times B_i^2 \qquad (1)$$

for every color in the palette, where ω_1, ω_2, and ω_3 are weighting values that make color such as (140,130,120) to be different from (130,140,120), and $\omega_1 \neq \omega_2 \neq \omega_3$. Then we sort $\mu(p_i)$ in a

descending order to obtain a sorted palette. Fig. 1 (a) shows a color palette before sorting and Fig. 1 (b) shows the sorted result of Fig. 1 (a).



(a) (b)

Figure 1. The color palette before and after sorting. (a) The color palette before sorting. (b) The color palette after sorting.

3. Data Hiding Scheme by Palette Index Replacement

In this section, the proposed method for embedding and extracting binary data information will be described. An even-odd relationship between two pixels is utilized to embed or extract a binary bit. The embedding process will be described in Section 3.1. And the extracting process will be described in Section 3.2. Some experimental results will be shown in Section 3.3.

3.1 Proposed Data Hiding Method

Let *C* be a cover image of size $M \times N$. Let *S* be the annotation that will be hidden into *C*, which has *L* characters in length. And let β be the sorted palette, with β_ι ($0 \le \iota \le 255$) being a color inside β that has index ι. Before the embedding process, *S* must first be converted into a binary form $S = s_1 s_2 \ldots\ldots s_8 \ldots\ldots s_{L*8-1} s_{L*8}$ according to the ASCII codes of the characters in *S*. As Fig. 2 shows, we hide the binary data into the four marked area (*I*, *II*, *III*, and *IV*) of every 3x3 block in *C*, and leave the central pixel unchanged, which will be used for hiding fragile information as described in Section 4.

Let the Euclidean distance between the two colors be

$$\mu(\beta_x, \beta_y) = \sqrt{(R_x - R_y)^2 + (G_x - G_y)^2 + (B_x - B_y)^2}, \quad (2)$$

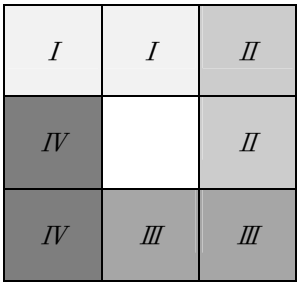


Figure 2. A 3×3 block.

where $0 \le x, y \le 255$ .The algorithm for embedding a bit of annotation, say s_i, can be expressed briefly as follows.

Step 1: Select a two-pixel block (*I*, *II*, *III* or *IV*) first from a 3x3 block. Name the two involved pixels as α_1 and α_2.

Step2: By viewing every pixel as an index number in the color-palette image, get the color indices ι_1 and ι_2 of the two selected pixels α_1 and α_2 from the sorted palette β, respectively.

Step3: Use the following rule to judge whether ι_1 and ι_2 are even or odd numbers:

$$isEven(\iota) = \begin{cases} true & \text{if } \iota \bmod 2 = 0; \\ false & \text{if } \iota \bmod 2 = 1. \end{cases} \quad (3)$$

Step4: In the case that $s_i = 1$ and $isEven(\iota_1) \neq isEven(\iota_2)$, modify one of the pixels by another color in β to make the two indices either both even or both odd in a quality preserving fashion. Let β_{r_1} and β_{r_2} be two candidate colors that will be used to replace α_1 or α_2, respectively. Choose the color β_{r_1} to be the one among β that has minimal Euclidean distance to α_1 and satisfy the condition $isEven(\iota_1) \neq isEven(r_1)$. On the other hand, choose the candidate color β_{r_2} to be the one among β that has minimal Euclidean distance to α_2 and satisfy the condition that

$isEven(\iota_2) \neq isEven(r_2)$. And then modify one of the pixels by following condition:

$$\begin{cases} if\ \ \mu(\alpha_1, \beta_{r_1}) > \mu(\alpha_2, \beta_{r_2}) & \text{replace } \alpha_2 \text{ by } \beta_{r_2}, \\ if\ \ \mu(\alpha_1, \beta_{r_1}) < \mu(\alpha_2, \beta_{r_2}) & \text{replace } \alpha_1 \text{ by } \beta_{r_1}. \end{cases} \quad (4)$$

Otherwise, in the case that $s_i = 1$ and $isEven(\iota_1) = isEven(\iota_2)$, just leave the two pixels unchanged.

Step5: In another case that $s_i = 0$ and $isEven(\iota_1) = isEven(\iota_2)$, modify one of the pixels to make one of the index be even and the other be odd. Perform similar operations in Step 4 to obtain two candidate colors β_{r_1} and β_{r2} from β and use them to replace α_1 or α_2 according to the condition listed in Eq. (4). Otherwise, if $s_i = 0$ and $isEven(\iota_1) \neq isEven(\iota_2)$, just leave the two pixels unchanged.

In the proposed method, the even-odd relationship of the two chosen pixels is utilized to embed a binary bit of annotation information. The indices ι_1 and ι_2 are tuned up to become both even or both odd to represent the fact that a "1" is embedded. On the contrary, a "0" is embedded by adjusting the indices of the two chosen pixels to become different in the even-odd relation, that is, for one index to become even and the other to become is odd. There is a good chance that the original indices just fit these two constraints. In this kind of situation, what we have to do is just to leave the two pixels unchanged. On the other hand, when the indices of the two pixels do not meet the constraints, a color with its index satisfying to the constraints described in Step 4 and Step 5 must be found to replace the corresponding pixel. In order to preserve the quality of the stego-image, the color with the minimal Euclidean distance to the corresponding pixel is the best candidate for replacement.

3.2 Data Extraction Process

In some conventional methods, besides the embedding result, an extraction process needs either a lookup table or the source image to obtain the secret message. This is inconvenient for the receiver to get the secret message because the user has to keep the source image or a table. In our proposed method, only the stego-image is needed to extract the annotation message.

Because the annotation embedded in the image can be any kind of description with an unfixed length, in our implementation the length of the annotation is first embedded before the annotation content is treated. As a result, the length information must first be extracted in the extraction process so that we can know how many blocks should be examined before ending the process.

Let *E* be the stego-image to be processed, β be the sorted palette of *E*, and s_i be the *i*-th extracted bit of the annotation data. The process of extracting a bit from the stego-image is described below:

Step1: Choose a two-pixel block (see Figure 2) from a 3×3 block of *E*. And name the two involved pixels α_1 and α_2.

Step2: Get the indices ι_1 and ι_2 of α_1 and α_2, respectively, from β according to their RGB values.

Step3: The annotation s_i can be determined by the following rule:

$$s_i = \begin{cases} 1 & \text{if } \iota_1 \text{ and } \iota_2 \text{ are both even} \\ & \text{or both odd;} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

In the entire extraction process, we first extract the first ℓ bits of the data that compose a number that is the value of the length of the annotation embedded in *E*. Then we can know how many two-pixel blocks we should examine and after repeating the above-mentioned process by $\ell \times 8$ times, we can get the binary form of the extracted annotation $S = s_1 s_2 \ldots\ldots s_8 \ldots\ldots s_{\ell*8-1} s_{\ell*8}$. According to the ASCII codes, *S* is then converted into character form that finally results in the embedded annotation.

3.3 Experimental Results

In our experiments, the images "Lena", "Baboon", and "Pepper" with size 512×512, which are shown in Fig. 3(a), (b), and (c), are used as the cover images. And Fig. 3(d), (e), and (f) are the stego-images after embedding 10,000 characters. The PSNR values of the stego-images are listed in Table 1. The PSNR values are high, which mean that only little perceptual effect is created during the annotation embedding process. And the embedded annotation can be extracted without any error, according to the experimental results.

Table 1. The PSNR values after embedding 10,000 characters.

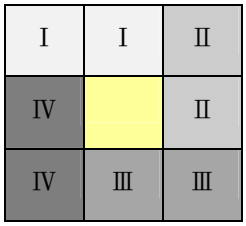
	Lena	Baboon	Pepper
PSNR	39.0	35.0	37.0

4. Authentication Scheme by Nearest Palette Color Replacement Method

In this section, a novel method for authenticating the integrity and fidelity of the color-palette image is proposed. Alternations to the watermarked image can be detected and localized. And the verification processes can be proceeded without referencing the original image. In Section 3, the secret data is embedded in the surrounding eight pixels of a 3×3 block and the central pixel is left unchanged. In the proposed method, the central pixel of every 3×3 block is utilized to embed the fragile watermark. The fragile watermark embedding process will be shown in Section 4.1. The authenticating process will be described in Section 4.2. And several experimental results are shown in Section 4.3.

4.1 Proposed Fragile Watermark Embedding Method

Let C be a cover image of size $M \times N$, β be the sorted palette of C, and β_i be the *i-th* color in β ($0 \le i \le 255$). The cover image C is first divided into non-overlapping 3×3 image blocks, like the one shown below.



We utilize the surrounding four two-pixel blocks (I, II, III and IV) to embed the secret data by using the method described in Section 3. After embedding, every block's central pixel is still unchanged. This pixel is hence used for embedding the fragile watermark. To embed the fragile watermark into a 3×3 block B, the algorithm can be shown as the following steps:

Step 1: Calculate the mean value μ of B, which can generally represent the color feature of a block. And the central pixel of the block is supposed to be close to μ in color. Since μ is the mean color of a block, there is a good chance that it is not contained in β.

Step2: Find a color β_k from β that is closest to μ according to the Euclidean distance of the RGB values, and $k_{\bmod Q} = 0$. Here Q is a watermark strength factor.

Step3: Replace the color value of central pixel of the block by β_k.

The magnitude of Q in Step 2 is a tradeoff between the probability to prove the tampering and the quality of the stego-image. When Q is a small value, the quality of stego-image is relatively higher because the candidate colors could be more so that we can find a closer color for replacing the central pixel of every 3×3 block. But the opportunity of proving and localizing the tampering will became relatively lower. On the contrary, if Q is large, the image quality will become poorer but the probability of detecting tampering will become higher. The detailed data about the influence of Q and some discussions will be shown in Section 4.3.

(a)

(d)

(b)

(e)

(c)

(f)

Figure 3. The cover images and stego-image with 10,000 characters embedded. (a) Cover image “Lena”. (b) Cover image “Baboon”. (c) Cover image “Pepper”. (d) Stego-image “Lena” with 10,000 characters embedded. (e) Stego-image “Baboon” with 10,000 characters embedded. (f) Stego-image “Pepper” with 10,000 characters embedded .

4.2 Authentication Process

The authentication process can proceed without referencing the original image in our proposed method. Let T be the image that is suspicious of being tampered. And let β be the sorted palette of T, and β_i be the i-th color in β ($0 \le i \le 255$). To authenticate whether T has been tampered, we first divide T into 3×3 non-overlapping blocks. For every block, we focus on the central pixel p and get the palette index k of p from β, where the RGB values of β_k are equal to those of p. In the embedding process described in Section 4.1, the central pixel of every block is replaced by the color whose index is a multiple of Q. Therefore, k should be a multiple of Q if this block has not been tampered with. That is, if k is a multiple of Q, the block is judged as not being tampered with. On the other hand, the block is decided as being tampered with if k is not a multiple of Q.

For visualization, we replace the block by its inverse color if the authentication result indicates the block has been altered. This will help us to localize the tampering area in the suspicious image.

4.3 Experimental Results

In our experiments, we use three cover images as shown in Fig. 4(a), (b), and (c) with size 512×512. And Fig. 4 (d), (e), (f) are the images after embedding a fragile watermark with Q = 8. We almost cannot tell the difference between the cover and the stego-images. The PSNR values are shown in Table 2.

Table 2. PSNR values of watermarked images.

	Jet	Lena	Pepper
PSNR	36.0	35.4	34.2

Figs. 5(a), (b), and (c) are the images that have been tampered. And the authentication results are shown in Figs. 5(d), (e), and (f). In Fig. 5(a), we tampered the stego-image by rubbing the characters and the flag on the body of the airplane. In Fig. 5(b), we altered Lena's hair color and paste some wig on her hat. In Fig. 5(c), we cropped two green peppers and pasted them onto different parts of the stego-image. The experimental results showed that the altered areas are accurately marked out.

Fig. 6 are the embedded results of "Lena" by applying different Q values. When Q becomes larger, the quality of the stego-image goes down. In Fig. 6 (c) and Fig. 6 (d), we can find some false contouring effect on the girl's shoulder and cheek because there are only 4 or 2 colors can be used for replacing the central pixel of every 3x3 block. The relation between the size of Q and the PSNR values of the stego-image is shown in Table 3.

Table 3. The PSNR values of stego-image by applying different Q values.

	Q=16	Q=32	Q=64	Q=128
PSNR	33.2	32.2	29.0	25.0

5. Discussions

In this thesis, a system that embeds annotation data and a fragile watermark simultaneously within the color-palette image has been proposed. The color-palette image contains a palette with all the colors used in that image. In order to reference the palette in the same order, the color palette is first reordered from dark to bright. In the embedding process, the annotations are embedded in the eight surrounding pixels of a 3×3 block. To embed a bit of data, two pixels are first selected and the indices of these two pixels are obtained from the color palette. The indices are tuned up to become both even or both odd if a "1" is to be embedded. Otherwise, the indices are tuned up to become different in the even-odd relation, that is, with one even and the other odd. To embed the fragile watermark, the mean value of every 3×3 block is calculated. And the central pixel is replaced by a color that is closest to the mean value of the block. And the magnitude of the index of this color must be a multiple times of the value of a predefined watermark strength factor.

(a)

(d)

(b)

(e)

(c)

(f)

Figure 4. The cover images and the fragile watermarked images (a) Cover image “Jet”. (b) Cover image “Lena”. (c) Cover image “Pepper”. (d) Watermarked image “Jet”. (e) Watermarked image “Lena”. (f) Watermarked image “Pepper”.

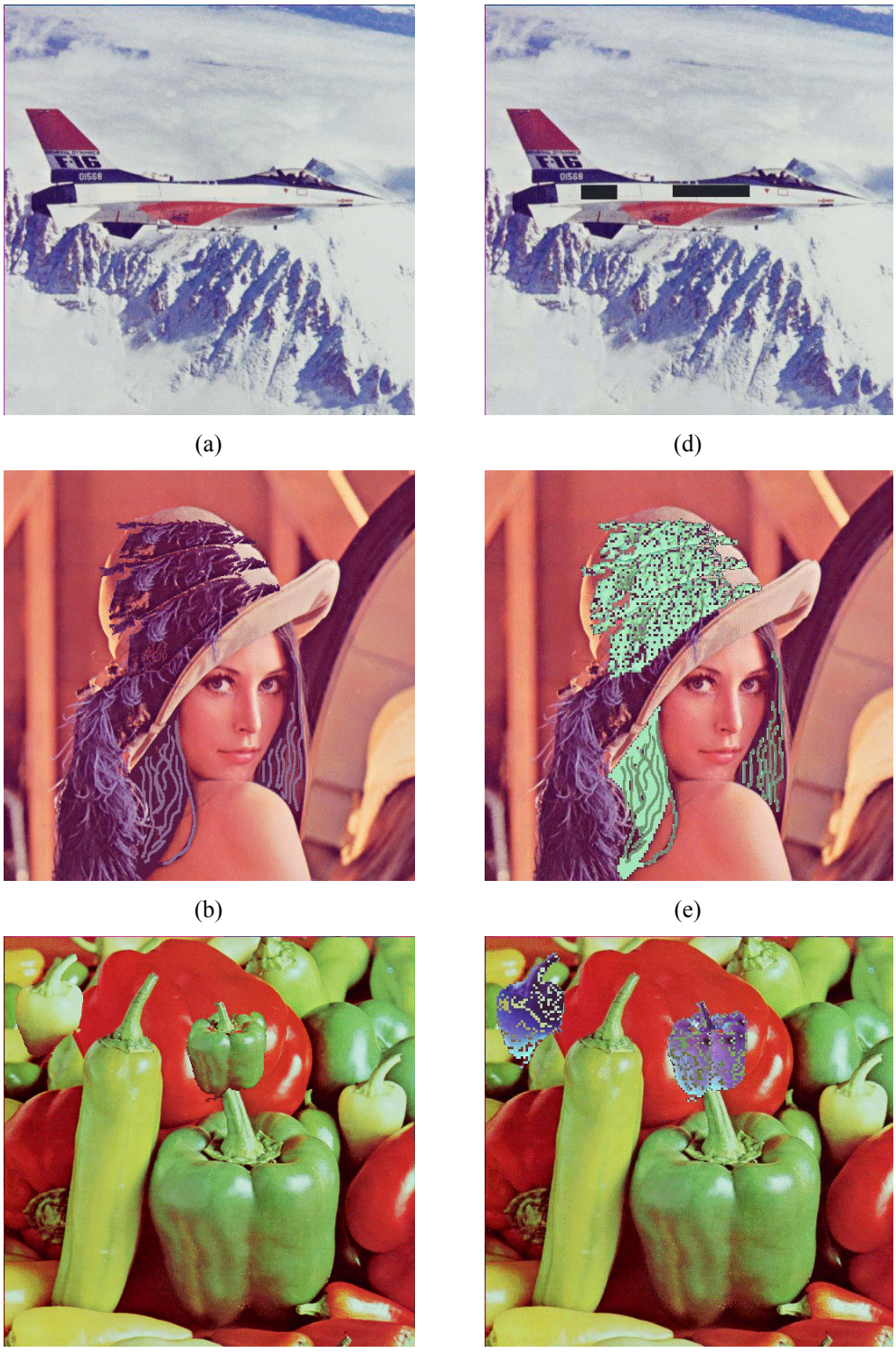


Figure 5. The suspicious images and their verification results. (a) Tampered “Jet”. (b) Tampered ”Lena”. (c) Tampered “Pepper”. (d) Verification result of tampered “Jet”. (e) Verification result of tampered “Lena”. (f) Verification result of tampered “Pepper”.

(a) Q = 16 (b) Q = 32

(c) Q = 64 (d) Q = 128

Figure 6. The watermarked image with different Q values. (a) Watermarked image with Q = 16. (b) Watermarked image with Q = 32. (c) Watermarked image with Q = 64. (d) Watermarked image with Q = 128.

References:

[1] D. C. Wu and W. H. Tsai, "Embedding of Any Type of Data in Images Based on a Human Visual Model and Multiple-Based Number Conversion," accepted and to appear in *Pattern Recognition Letters*.

[2] D. C. Wu and W. H. Tsai, "A Method for Creating Perceptually Based Fragile Watermarks for Digital Image Verification," submitted to *IEEE Transactions on Multimedia*.

[3] D. Kundur and D. Hanzinakos, "Towards a Tell-Tale Watermarking Technique for Tamper-Proofing," *in Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 409-413, Chicago, Illinois, 1998.

[4] D. Stinson, *Cryptography Theory and Practice*, CRC Press, Boca Raton, 1995.

[5] E. H. Adelson, "Digital signal encoding and decoding apparatus," U.S. Patent 4939515,1990.

[6] H. Y. Chang, "Data hiding and watermarking in color images by wavelet transforms," Master thesis, Department of Computer and Information Science, National Chiao Tung University, Taiwan, Republic of China, 1999.

[7] J. Fridrich, "Image watermarking for tamper detection," in *Proc. IEEE International Conference on Image Processing*, vol. II, 1998, pp. 404-408.

[8] M. S. Liaw and L. H. Chen, "An effective data hiding method," in *Proc. IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Taiwan, R.O.C., 1997, pp.146-153.

[9] M. Wu and B. Liu, “Watermarking for image authentication,” *in Proc. IEEE International Conference on Image Processing*, vol. II, pp. 437-441, Chicago, Illinois, October 1998.

[10] P. W. Wong, “A public key watermark for image verification and authentication,” in *Proc. IEEE International Conference on Image Processing*, vol. II, 1998, pp. 455-459.

[11] R. van Schyndel, A. Tirkel, and C. Osborne, “A Digital Watermark,” *Proceeding of the IEEE International Conference on Image Processing*, vol 2, pp. 86-90, Austion, Texas, November 1994.

[12] S. Walton, “Information authentication for a slippery new age,” Dr. Dobbs Journal, vol. 20, no. 4, pp. 18-26, April 1995.

[13] T. K. Yen, “Image hiding by random bit replacement and frequency transformations,” *Master thesis, Department of Computer and Information Science, National Chiao Tung University, Taiwan, Republic of China*, 1998.

[14] T. S. Chen, C. C. Chang, and M. S. Hwang, “A virtual image cryptosystem based upon vector quantization,” *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1485-1488, 1998.