Efficient Steganographic Embedding by Exploiting Modification Direction

Xinpeng Zhang and Shuozhong Wang

***Abstract*— A novel method of steganogrphic embedding in digital images is described, in which each secret digit in a $(2n+1)$-ary notational system is carried by n cover pixels and, at most, only one pixel is increased or decreased by 1. In other words, the $(2n+1)$ different ways of modification to the cover pixels correspond to $(2n+1)$ possible values of a secret digit. Because the directions of modification are fully exploited, the proposed method provides high embedding efficiency that is better than previous techniques.**

***Index Terms*— Covert communication, steganography, embedding efficiency, embedding rate.**

I. Introduction

DIGITAL steganography is a technique for sending secret messages under the cover of a carrier signal. Despite that steganographic techniques only alter the most insignificant components, they inevitably leave detectable traces so that successful attacks are often possible. The primary goal of attack on steganographic systems, termed steganalysis, is to detect the presence of hidden data by finding statistical abnormality of a stego-media caused by data embedding [1], [2]. Generally speaking, the more the secret data are embedded, the more vulnerable is the steganographic system to steganalytic attempts.

One way of improving security of a steganographic system is to reduce the amount of alterations necessary to be introduced into the cover signal for data hiding when the number of secret bits is significantly less than that of available host samples. For example, the method proposed in [3] can conceal as many as $\lfloor \log_2 (mn+1) \rfloor$ bits of data in a binary image block sized $m \times n$ by changing, at most, two bits in the block. Matrix encoding [4], on the other hand, uses less than one change of the least significant bit (LSB) in average to embed l bits into $2^l - 1$ pixels. In this way, the introduced distortion is significantly lowered compared to a plain LSB replacement technique in which secret bits are used to simply replace the LSB plane. Further, some effective encoding methods derived from the cyclic coding have been described [5], and the matrix encoding can be viewed as a special case. In [6], two steganographic encoding methods based on random linear codes and simplex codes are developed for large payloads. The stego-coding technique can also be performed on a data stream derived from the host in a dynamically running manner, so that insertion and extraction of each secret bit are carried out in a series of consecutive cover bits [7]. All the above-mentioned stego-coding techniques are independent of various cover-bit-modification approaches. For example, if the stego-coding methods are used in the LSB plane of an image, adding 1 to a pixel is equivalent to subtracting 1 from the pixel to flip its LSB for carrying the secret message.

In [8], another type of steganographic method for improving embedding efficiency is presented in which the choice of whether to add or subtract one to/from a pixel depends both on the original gray value and a pair of two consecutive secret bits. In other words, the direction of modification to the cover pixels is exploited for data hiding. However, there exist two different modification-directions corresponding to a same pair of secret bits to be embedded, meaning that the exploitation is incomplete.

This letter proposes a novel steganographic embedding method that fully exploits the modification directions, called the EMD embedding for short. In this method, modifications in different directions are used to represent different secret data, leading to a higher embedding efficiency.

II. EMD Embedding

The main idea of the proposed steganographic method is that each secret digit in a $(2n+1)$-ary notational system is carried by n cover pixels, where n is a system parameter, and, at most, only one pixel is increased or decreased by 1. Actually, for each group of n pixels, there are $2n$ possible ways of modification. The $2n$ different ways of alteration plus the case in which no pixel is changed form $(2n+1)$ different values of a secret digit.

Before data-embedding, a data-hider can conveniently convert a secret message into a sequence of digits in the notational system with an odd base $(2n+1)$. If the secret message is a binary stream, it can be segmented into many pieces with L bits, and the decimal value of each secret piece is represented by K digits in a $(2n+1)$-ary notational system, where

$$L = \lfloor K \cdot \log_2 (2n+1) \rfloor \tag{1}$$

For example, the binary sequence (1101 0110 1001) can be expressed as (23 11 14) in a 5-ary notational system where $L = 4$ and $K = 2$. Thus, the redundancy rate in the $(2n+1)$-ary sequence is

$$R_R = 1 - \frac{L}{K \log_2 (2n+1)} < \frac{1}{L+1} \tag{2}$$

Manuscript received June 5, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Deepa Kundur. This work was supported in part by the National Natural Science Foundation of China under Grants 60372090 and 60502039, in part by the Key Project of Shanghai Municipality for Basic Research under Grant 04JC14037, and in part by Shanghai Leading Academic Discipline Project under Grant T0102.

The authors are with the School of Communication and Information Engineering, Shanghai University, Shanghai 200072 China (e-mail: {xzhang, shuowang}@staff.shu.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2006.060863.

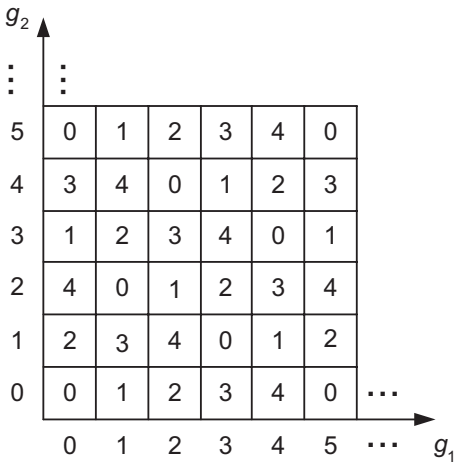

Fig. 1. Extraction function of 2D hyper-cubes (squares), each of which corresponds to a pair of gray-values.

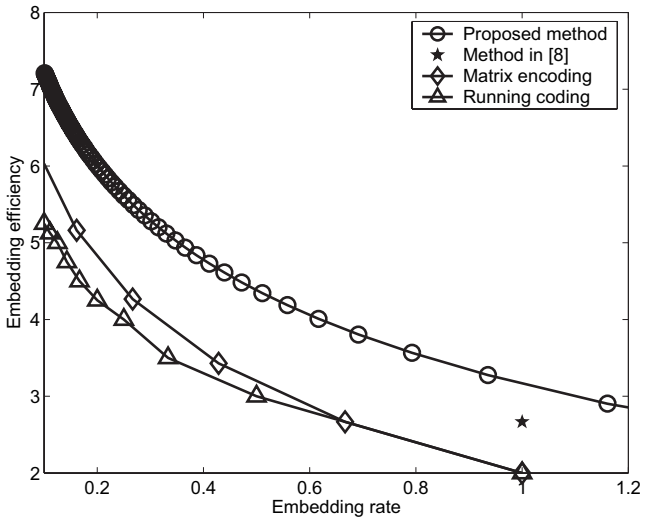

Fig. 2. Performance comparison among the proposed method, running coding, matrix encoding, and the method in [8]. The abscissa and the ordinate represent embedding rate and embedding efficiency respectively.

With large L and K, R_R is very close to 0, therefore can be ignored. So, the secret message is regarded as a sequence of digits in the $(2n+1)$-ary notational system.

Let us embed data in an uncompressed cover image. With the proposed method, n pixels are used to carry one secret digit in the $(2n+1)$-ary notational system and, at most, only one pixel is increased or decreased by 1. Pseudo-randomly permute all cover pixels according to a secret key, and divide them into a series of pixel-groups, each containing n pixels. Denote the gray values of pixels in a group as $g_1, g_2, \ldots, g_n$, and calculate the extraction function f as a weighted sum modulo $(2n+1)$:

$$f(g_1, g_2, \ldots, g_n) = \left[\sum_{i=1}^{n} (g_i \cdot i)\right] \bmod (2n+1) \quad (3)$$

Since gray values are integers, a vector $[g_1, g_2, \ldots, g_n]$ in the n-dimensional space may be represented by a unit hyper-cube as shown in Fig. 1 for the simplest case of $n = 2$, in which each hyper-cube (square) is labeled with its f value. The f values of any hyper-cube and its $2n$ neighbors, which are integers within $[0, 2n]$, are mutually different.

Map each secret digit in the $(2n+1)$-ary notational system to a pixel-group. No modification is needed if a secret digit d equals the extraction function of the original pixel-group. When $d \neq f$, calculate $s = d - f \bmod (2n+1)$. If s is no more than n, increase the value of g_s by 1, otherwise, decrease the value of g_{2n+1-s} by 1. For example, consider an original pixel-group [137 139 141 140] with $n = 4, f = 3$ and a corresponding secret digit 4 in a 9-ary notational system. Since $s = 1$, a data-hider would increase the gray value of the first pixel by 1 to produce the stego-pixels [138 139 141 140]. If the secret digit to be hidden is 0, $s = 6$ can be calculated and the gray value of the third pixel will be decrease by 1 to yield [137 139 140 140]. On the receiving side, the secret digit can be easily extracted by calculating the extraction function of stego-pixel-group. In fact, any n-dimensional hyper-cube and its $2n$ neighbors are used to represent the different values of a secret digit, leading to a complete exploitation of modification-direction.

However, increase of g_s or decrease of g_{2n+1-s} may not be allowed if the pixel is saturated. To deal with this problem, change the saturated pixel by 1 and embed the secret digit again. As an example, consider a pixel-group [255 255 255 254] with $n = 4$ and $f = 8$. For a secret digit 0, $s = 1$. This requires an increase of the first pixel that is not allowed. In this case, decrease the first pixel by 1 to make the group [254 255 255 254] with $f = 7$ so that $s = 2$, which is still prohibitive since the second pixel is also saturated. Again, make the group [254 254 255 254] with $f = 5$. This time the digit 0 can be embedded successfully to give the stego-group [254 254 255 255]. The same extraction method can be used to retrieve the embedded digit. Although saturation may require modifications of more than one pixel, it does not affect the overall performance since saturated pixels of a natural image are rare.

Two parameters are used as the performance metrics: embedding efficiency E that is a ratio between the number of embedded bits and the distortion energy caused by data embedding, and embedding rate R that is the number of secret bits embedded in each cover pixel. In the proposed method, a digit in the $(2n+1)$-ary notational system, representing $\log_2(2n+1)$ bits, is embedded into n pixels, among which one pixel is increased or decreased by 1 with a probability $2n/(2n+1)$, thus,

$$E = \frac{(2n+1) \cdot \log_2(2n+1)}{2n} \quad (4)$$

and

$$R = \frac{\log_2(2n+1)}{n} \quad (5)$$

Performance comparison has been made among the proposed EMD method, the running coding [7], the matrix encoding [4], and the method described in [8] as shown in Fig. 2. The method of [8] incurs less changes in the host image

compared to a plain LSB matching technique, therefore has improved security against the HCF-COM analysis introduced in [9] and developed in [10]. The results in Fig. 2 show that, at any given embedding rate, the proposed method has the highest embedding efficiency, resulting in the least distortion and best security.

III. Discussion

It has been shown that the matrix encoding method is derived from the binary Hamming coding [6]. The author of [5] independently presents the same steganographic technique and additionally discusses a similar stego-encoding method derived from the ternary Hamming coding. By ternary matrix encoding, the secret data with $\log_2 3^t$ bits are carried by $(3^t - 1)/2$ cover pixels, where t is a positive integer, and no more than one pixel is increased or decreased by 1 for data hiding. Actually, the proposed EMD embedding scheme in cases of $n = (3^t - 1)/2$ is equivalent to ternary matrix encoding. On the other hand, the EMD method provides more selectable cases so that the secret message can be conveniently hidden irrespective of the ratio between the payload and the size of cover signal.

References

[1] H. Wang and S. Wang, "Cyber warfare: steganography vs. steganalysis," *Communication of the ACM*, vol. 47, no. 10, pp. 76-82, 2004.

[2] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, "Quantitative steganalysis of digital images: estimating the secret message length," *Multimedia Systems*, vol. 9, no. 3, pp. 288-302, 2003.

[3] Y.-C. Tseng, Y.-Y. Chen, and H.-K. Pan, "A secure data hiding scheme for binary images," *IEEE Trans. Commun.*, vol. 50, no. 8, pp. 1227-1231, Aug. 2002.

[4] A. Westfeld, "F5: a steganographic algorithm," in *Proc. 4th Int. Workshop Information Hiding 2001, Lecture Notes in Computer Science*, vol. 2137, pp. 289-302.

[5] M. Dijk and F. Willems, "Embedding information in grayscale images," in *Proc. 22nd Symp. Information Theory Benelux 2001*, pp. 147-154.

[6] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," in *Security, Steganography, and Watermarking of Multimedia Contents VIII (2006), Proc. SPIE*, vol. 6072, pp. 60721W1-12.

[7] X. Zhang and S. Wang, "Dynamically running coding in digital steganography," *IEEE Signal Processing Lett.*, vol. 13, no.3, pp. 165-168, Mar. 2006.

[8] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing Lett.*, vol. 13, no. 5, pp. 285-287, May 2006.

[9] J. Harmsen and W. Pearlman, "Steganalysis of additive-noise modelable information hiding," in *Proc. SPIE Security Watermarking Multimedia Contents 2003*, vol. 5020, pp. 131-142.

[10] A. Ker, "Steganalysis of LSB matching in greyscale images," *IEEE Signal Processing Lett.*, vol. 12, no. 6, pp. 441-444, June 2005.