A Low Complexity Embedding Algorithm Based on a Low Weight Search Method for Steganography

Zih-Syuan Lian [#1], Chi-Yuan Lin [#2], Jyun-Jie Wang [*3], Houshou Chen [!4]

[#]*Department of computer science and information engineering, National Chin-Yi University of Technology*

No.57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan (*R.O.C.*)

[1]esthersky1119@gmail.com

[2]chiyuan@ncut.edu.tw

[*]*Department of Electronic Engineering, National Chin-Yi University of Technology*

No.57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan (*R.O.C.*)

[3]d9464108@mail.nchu.edu.tw

[!]*Department of Electrical Engineering, National Chung Hsing University*

250 KuoKuang Rd., Taichung 402, Taiwan (*R.O.C.*)

[4]houshou@dragon.nchu.edu.tw

***Abstract*—This study proposes a novel suboptimal hiding algorithm for binary data based on lowweight search embedding (LWSE). With a constant distortion, this algorithm exhibits the advantage of a fast embedding algorithm for halftone images. The suboptimal LWSE algorithm performs an embedding procedure through a fast algorithm using a parity check matrix. Unlike the maximal likelihood (ML) algorithm, the purpose is to locate a coset leader. Using the proposed method, locating a coset leader is unnecessary. Instead of locating the coset leader, this algorithm uses a LWSEmethod attempt to locate the low-weight vector corresponding to the toggle syndrome. In addition to this leading to an improved embedding efficiency, this paper presents an expanded embedding code. With a linear embedding code, in the case of an embedding complexity, the Order 2 to the power of k, using the optimal ML algorithm, the operation complexity required in the suboptimal LWSE is of a lower complexity $O(\lambda k)$.**

***Keywords*—Embedding efficiency, embedding rate, embedding speed, matrix embedding, steganography.**

1. INTRODUCTION

Steganography must be realized by numerous reliable methods, one of which is the so called matrix embedding (ME) [1]. ME refers to the technique embedding data into a host vector or a cover object, such as images, videos, or audio. Their applications are varied, such as copyright protection, that is, watermarking, steganography and content authentication.

Binary data embedding had been implemented using the suboptimal embedding algorithm in numerous research studies [2~10]. In 2006, Fridrich et al. [2] showed that the security of steganography for large payloads can be improved by simplex codes and random codes. However, the computational complexity in [2] is high, and cannot reach real-time applications. In 2007, Liet al. proposed that a steganographic algorithm is a tree-based parity check (TBPC) algorithm [4], which is a suboptimal embedding algorithm characterized by a tree structure. Although the TBPC algorithm is simple, the embedding efficiency for the steganographic scheme is poor. However, the MPC algorithm, proposed by [8] further improves the embedding efficiency of the TBPC algorithm. The MPC method can be formulated as ME, and optimize the ME by using tree structure. The MPC method provides efficient embedding algorithms. Although the MPC method is an efficient algorithm, its embedding efficiency remains poor. [10] proposed an ME method, which the number of referential columns to theparity check matrix, to reduce the computationalcomplexity of random linear code, and to increase the embedding efficiency of the steganographic scheme. [10] use random linear code based matrix, which was extended, to achieve high embedding efficiency; however, it was expensive to compute. Howto reduce the computational complexity of the algorithm and steganographic scheme while maintaining high embedding

efficiency is a critical problem. However, those binary embedding algorithms are special cases of linear embedding codes, because an arbitrary parity check matrix is capable of embedding binary data. To reduce the embedding complexity, ME is an efficient method for increasing embedding efficiency.

This paper considers the trade-off between embedding efficiency and computational complexity. It proposes a fast and low complexity LWSE algorithm for the parity check matrix, called low-weight search embedding (LWSE). LWSE has the advantage of high embedding efficiency for small payloads. Another advantage of LWSE is that its implementation is suitable for various embedding rates. An important goal for the embedding algorithm can be implemented with linear computation complexity. Embedding algorithms based on ME can improve embedding efficiency. Moreover, the steganographic scheme based on the parity check matrix can be extracted in the receiver by a multiplication operation occurring between the parity check matrix and received setgo. In principle, the maximum embedding rate can be estimated through the binning methods when the embedding data are the binary source. Furthermore, with an existing parity check code, the embedding rate, which corresponds to a minimum level of distortion, can be determined accordingly. By the time an embedding scheme is built that reaches this type of limit, two concerns are raised.

1) With reference to a linear block code, a parity check code requires a well-performing parity check matrix.
2) The most efficient embedding algorithm is found on the basis of this coding structure, when the first requirement is met.

The rest of this paper is organized as follows. In Section 2, we proposed suboptimal embedding algorithm. In Section 3 the embedding complexities of the algorithms presented here. In Section 4, we provide experimental results. Finally, we presented our conclusions in Section 5.

2. SUBOPTIMAL EMBEDDING ALGORITHM

For an embedding scheme, with binary toggle vector $x_{opt} \in \{0,1\}^n$, the output of the embedder is $l_{opt} = u - x_{opt}$, where x_{opt} is provided by an embedder, that is, a stego l_{opt} modified from u and corresponding to the message s_l. The scheme of an embedding strategy can be realized by linear codes; an (n,k) linear code C at an embedding rate of $R_e = m/n$, and supposing that the toggle vector x_{opt} must satisfy the constraint $E[w_H(x_{opt})] \le \delta n$, where $w_H(x_{opt})$ denotes the Hamming weight and $0 \le \delta \le 0.5$. Assume that the linear codes C at embedding rate R_e correspond to an embedding average distortion of $d_{avg} = E[w_H(x_{opt})]/n$. The theoretically achievable bound is derived as $h(d_{avg}) \ge R_e$, where $h(d_{avg})$ denotes a binary entropy function. Thus, a bound on the maximal embedding efficiency η can be obtained as

$$\eta \le \frac{R_e}{h^{-1}(R_e)}.$$

An efficient algorithm is presented here toper form binary data embedding. As mentioned, it is unrealistic to build a standard array corresponding to an arbitrary large linear block code. Here, the coset leader is found in an alternative manner to the conventional maximum likelihood (ML) decoding algorithm. A simple method is to locate a low-weighted toggle vector during the search of coset leader vectors e_{opt}. It is intended to locate vector e_{sub} and $w_H(e_{sub}) \ge w_H(e_{opt})$, in lieu of the optimal coset leader $w_H(e_{opt})$. The e_{sub} should remain as near $w_H(e_{opt})$ as possible. The e_{sub} is a vector defined by C^x. Obtained by the addition of e_{sub} to the host vector u, the target vector l' cannot be ensured as the optimal vector.

2.1. Suboptimal binary embedding

Although the linear block code for an embedding algorithm is simple and has low complexity, this paper further proposes an embedding algorithm based on the parity check matrix, which not only has low complexity but also performs the algorithm quickly. Numerous feasible binary data embedding algorithms are presented. The suboptimal algorithm, that is, the LWSE algorithm, is stated. Unlike the ML embedding algorithm, a minimum weighted toggle vector is searched in this iterative technique. Specifically, in the ML algorithm, the minimum weight is obtained through an entire search of codewords within the linear embedding

code; however, the LWSE algorithm seeks the less-weighted toggle vector by times of iterations. Performing k times of hardware operations each time , LWSE proceeds in an iterative manner until convergence is reached. The LWSE algorithm provides a lower operation complexity than the ML algorithm does at the cost of distortion efficiency.

2.2. Low- Weight Search Embedding Algorithm

The cover sequence is used to carry logo sequence in ME which is based on the parity check matrix. Assume that generating a binary matrix H of size $m \times n$ composed of two parts.

$$H = [IH_P]$$

The toggle sequence x is divided into two parts as $x = (x_M, x_P)$, where $x_M = x_1,\ldots,x_m$ and $x_P = x_{m+1},\ldots,x_n$. To locate the low weight of toggle sequence x , toggle sequence x must meet

$$s_x = Hx^T = x_M + H_P x_P^T \ldots\ldots\ldots(1)$$

toggle x of low weight, which is also represented as

$$x' = \arg\min(w_H(x_M) + w_H(x_P)) \ldots\ldots\ldots(2)$$

To minimize the weight of x' , (2) must simultaneously minimize the weight of x_M and x_P . However, this is a decoding problem using ML decoding. For ML decoding, the decoding complexity is 2^k . To solve (2) with a feasible computational complexity, [2] proposed using random linear codes (n,k) with a small dimension k .

This paper proposes an algorithm called LWSE, to obtain a toggle sequence with low weight. The LWSE algorithm uses two steps to minimize (2). The first is to minimize x_P . By using (1), let x_P be all zeros that meet the requirement; x_M is equal to s_x . The second step is to minimize $x_M = s_x$ based on the part of the parity check matrix $H_P = [h_{m+1}\ldots h_n]$, which forms column vectors h_i of systematic matrix H_p . By using (1), an x_P sequence is chosen which has one in the i th position, that is. To satisfy (1), column h_i corresponding to x_P out of linear code H_P is added to toggle $x_M = s_x$ as x'_M , that is, $x'_M = x_M + h_i$. Consequently, the weight of x'_M is altered through the columns of H_P , but x'_M still falls within coset C^x , $H(x'_M, x_P)^T = s_x$. These column vectors are defined as $H_P = [h_{m+1}\ldots h_n]$. For an arbitrary $h_i \in H_P$ toggle syndrome, s_x is known to be expressed as

$$\begin{aligned} & H(x'_M, x_P)^T \\ &= Ix'_M + H_P x_P \\ &= (x_M + h_i) + h_i \\ &= x_M \end{aligned}$$

Although 2^k columns are in the linear code C , it is unrealistic to choose all of the columns. Only k number of columns is selected from among the 2^k column sets. Although syndrome s_x remains invariant with h_i added to x_M , the toggle vector $x' = (x'_M, x_P)$ changes with the corresponding weight. The distortion is reduced in the event that a less-weighted modified toggle vector x' , is found that is lower than the original toggle vector x . Eventually, the distortion can be improved by means of a small amount of weight variation.The dimension of candidate H_P can be extended, and k number of column vectors can be selected out of H_p , or $H_P = \{h_i | i = 1\ldots k\}$ can be formed as a combination of two arbitrary vectors within H_p . However, the price paid for an increment in i is of a higher operational complexity. The case for $i = 1$ is addressed. Modified toggle vector $x' = (x_M + h_i, x_P)$ is gained through an appropriate weight variation of toggle x , with the primary goal of obtaining the weight of x' to that closest to the coset leader. Assume that $w_H(x) = \lambda$ and λ are constant, and vector x' approaching x can be expressed as

$$x' = \arg \min_{h_i \in H_p} w_H(x_M + h_i) + w_H(x_P)$$

where vector x' represents the vector with the minimum weight after k times of tests. In addition, in case $w_H(e) \le w_H(x') \le \lambda$ then vector x' remains closer to e than vector x . This algorithm, designated as LWSE, is capable ofreducing the toggle weight as much as possible in an iterativemanner.

It is unrealistic to perform ML decoding on the latter, because the decoding complexity increase exponentially with information bits $k = n - m$ as $O(2^k)$. A small increases in k leads to a dramatic

rise in complexity, causing ML decoding to be impossible in reality.

3. EMBEDDING COMPLEXITY

An (n,m) random linear code suffers the greatest disadvantage in embedding complexity in a high $n-m$ dimensional space. [2] proposed a means to improve embedding efficiency based on ME for large payloads using simplex codes. If the embedding rate is high, the dimension of codes is sufficiently small to enable fast decoding. The coset leader can also be found in ML decoding. If the dimension of codes is large, ML decoding, that is, the optimal embedding algorithm, is unrealistic. Numerous suboptimal embedding algorithms are presented, such as [4] and [8], [10]. These suboptimal algorithms can significantly reduce the embedding complexity as opposed to the ML embedding algorithm, at the expense of embedding efficiency. Specifically, the complexity of [4] can be up to $O(m)$ or $O(m\log_N m)$, and [8] is required to locate the majority vote value, which results in time complexity. Nevertheless, the LWSE suboptimal embedding algorithm merely requires a complexity of $O(\mu(n-m))$, associated with the iteration times. As stated in [2], a (n,m) random linear code is expanded as a means to gain an improved embedding efficiency for a large length, and it is enabled to perform the LWSE algorithm in a high $n-m$ dimensional space. Table 1 shows the speed and operation of embedding for various suboptimal embedding algorithms with fixe10^5random messages. [2] proposed two ME methods based on random linear codes and simplex codes. The time complexity of embedding algorithms for matrix embedding is bounded by the complexity of the decoding algorithms for the codes; that is, the complexity of finding the coset leader. The decoding algorithms for (n,k) simplex codes in [2] have a time complexity of $O(n\log n)$, where n is the code length and k is the dimension of the code. Although the embedding efficiency of [2] is near the upper bound for large payloads, the time complexity is still prohibitive. [4] and [8] proposed suboptimal embedding algorithms to improve the time complexity. [4] proposed a scheme to reduce embedding distortion based on a tree structure. This method is represented as ME, which is improved by Hou et al. [8] with the majority-vote parity check (MPC). Although [4] and [8] offer fast time complexity, the embedding efficiency of these algorithms is poor. [10] proposed a fast method for steganography by appending columns to the parity check matrix. This method can significantly reduce the computational complexity compared with the existing methods of ME using random codes [2] . In [10], this extended matrix is used to increase the embedding efficiency by using ML decoding and the computational complexity of [10] to exponentially increase from $O(n2^k)$ to $O(n2^{k+h})$. By using the proposed algorithm in [10], it must only search for a small solution space with a small size of the extended matrix and then the computational complexity is equal to $O((n+h)2^k)$, which linearly increases with h; thus this method has a faster embedding speed compared with the original ME [2]. An LWSE method is proposed to achieve an embedding algorithm with low complexity. Random codes (n,k) and Hamming codes were used to embed the fixed logo symbols. Because the proposed algorithm uses (n,k) codes, which are $(n-k)k$ in size, the memory requires $(n-k)kn$ in storage. Thus, the order of computations is $O(\lambda k)$. However, to keep the complexity and memory requirement slow, the amount of toggle vectors in the LWSE is low. Because the LWSE searches only a few search column vectors, the computational complexity requires that $O(\lambda k)$, where λ is constant.

TABLE 1
COMPARISON OF EMBEDDING THE TIME FOR FIXED 10^5 RANDOM MESSAGES WITH FIXED RELATIVE PAYLOAD $\alpha = 0.26$ AND BLOCK LENGTH

		Embedding time(10^3s , in matlab)	Embedding efficiency
(n , k) LWSE(hamming)	(15,11)	0.1814	4.2609
(n , k) LWSE(random)	(30,22)	0.0592	3.5542
[2] (n , k) Random	(19,14)	12.1360	3.3296
[10] (n , k , H) Me by me	(15,10,4)	2.0898	3.1791

4. SIMULATION RESULTS

For computational complexity concerns, the LWSE algorithm has a faster computational time compared to other embedding algorithms, as shown in Table1 . The mentioned algorithms are

simulated for their embedding efficiency. In the simulation, the cover and logo sequence is selected randomly. The cover is divided into the non-overlapping blocks; each block embeds the logo sequence of m bits. According to this, the LWSE algorithm, compared with the ML algorithm, requires a lower operation complexity, at the cost of degraded efficiency. The embedding efficiency corresponding to various systematic linear block codes is comparable between both the LWSE and numerous suboptimal algorithms. The proposed method can embed data for linear codes with large dimensions, that is, embedding for a small payload. Moreover, the LWSE algorithm can also embed data at various embedding rates. Considering the low rate in Fig. 1, the LWSE algorithm still follows the upper bound for embedding efficiency.

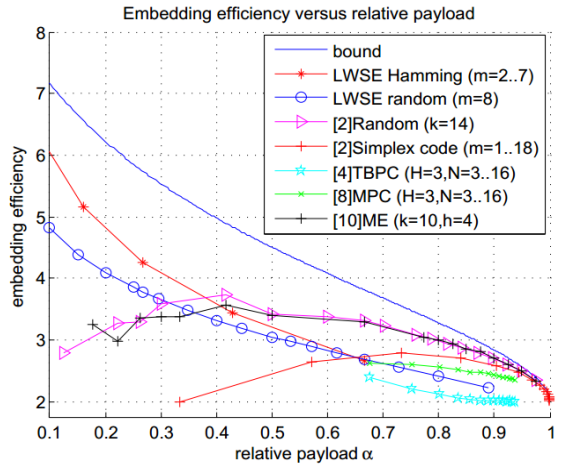

Fig. 1 Embedding efficiency of various algorithms.

5. CONCLUSIONS

This paper proposes a suboptimal embedding algorithm an LWSE algorithm, to reduce embedding complexity. Although the proposed scheme has decreased embedding efficiency compared to ME using ML decoding, it is can work suitable for various embedding rates and reduces the computational complexity. Moreover, the proposed algorithm has good embedding efficiency for embedding in small payloads compared to others embedding algorithms, as shown in Table 1. In the experiment, random codes and Hamming codes were used to implement the LWSE algorithm. Although LWSE caused some loss of embedding efficiency because ME used a suboptimal algorithm, the experimental results confirm that the LWSE algorithm is of low computational complexity compared with other embedding schemes.

ACKNOWLEDGMENT

The work was partially supported by the National Science Council of Taiwan, Republic of China under research contract NSC-101-2221-E-167-026.

REFERENCES

[1] R.Crandall, "Some notes on steganography," *Steganography Mailing List [Online]. Available: http://os.inf.tu-dresden.de/westfeld /crandall.pdf, 1998.*

[2] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Inf. Theory*, vol. 1, no. 3, pp. 390–395, Sep. 2006.

[3] R. Y. M. Li, O. C. Au, C. K. M. Yuk, S. K. Yip, and S. Y. Lam, "Halftone Image Data Hiding with Block-Overlapping Parity Check,'' *Proc. IEEE*, vol. 2, pp. 193-196, Apr. 2007.

[4] R. Y. M. Li, O. C. Au, K. K. Lai, C. K. M. Yuk, and S. Y. Lam, "DataHiding with Tree Based Parity Check," *IEEE International Conference*,pp. 635–638, Jul, 2007.

[5] Z. X. Qian, X. P. Zhang and S. Z. Wang, "Matrix Selection in High Payload Embedding," *in Proc. IIH-MSP2009*, pp. 328-331, Sep. 2009.

[6] Y. K. Gao, X. L. Li and B. Yang, "Employing optimal matrix for efficient matrix embedding," in *Proc. IIH-MSP2009*, pp.161-165, Sep. 2009.

[7] J. Y. Chen, Y. F. Zhu, Y. Shen and W. M. Zhang, "Efficient Matrix Embedding Based on Random Linear Codes," in *Proc.MINES2010*, pp. 879-883, Dec. 2010.

[8] C. Hou, C. Lu, S. Tsai, and W. Tzeng, "An optimal data hiding scheme with tree-based parity check," *IEEE Trans. Image Process.*, vol. 20, no.3, pp. 880–886, Mar. 2011.

[9] W. Y. Hung, C. Y. Lin, J. J. Wang, and P. C. Kuo, "A novel secret sharing scheme using forward error correction codes for halftone image," *The First National Conference on Web Intelligence and Applications*, Apr. 2011, vol. 1, pp. 545-551.

[10] C. Wang,W. Zhang, J. Liu, and N. Yu,"Fast Matrix Embedding by Matrix Extending,"*IEEE Trans. Inf. Theory*, vo7. 1, no. 1, pp. 346–350, Feb. 2012.