A Suboptimal Embedding Algorithm With Low Complexity for Binary Data Hiding

Jyun-Jie Wang and Houshou Chen
Graduate Institute of Communication Engineering and Dept. of Electrical Engineering
National Chung Hsing University
250, Kuo Kuang Rd., Taichung 402, Taiwan
E-mail: houshou@dragon.nchu.edu.tw

***Abstract*— A novel suboptimal hiding algorithm for binary data based on weight approximation embedding, WAE, is proposed. Given a specified embedding rate, this algorithm exhibits an advantage of efficient binary embedding with reduced embedding complexity. The suboptimal WAE algorithm performs an embedding procedure through a parity check matrix. The optimal embedding based on maximal likelihood algorithm aims to locate the coset leader to minimize the embedding distortion. On the contrary, the WAE algorithm looks for a target vector close to the coset leader in an efficiently iterative manner. Given an linear embedding code $C(n,k)$, the embedding complexity using the optimal algorithm is $O(2^k)$, while the complexity in the suboptimal WAE is reduced to $O(sk)$ where s is the average iterations.**

I. Introduction

As the demand of the public network communication increases significantly and a large amount of digital data must be transmitted in a number of secret ways, the technique of data hiding [1] becomes an important research area. Simply speaking, data hiding refers to the technique embedding data into a cover object, e.g., image, video, or audio, etc. The applications of data hiding, watermarking and steganography, are found in many aspects, such as copyright protection and content authentication. Besides, more concerns are raised in data hiding technique, such as capacity, distortion, robustness, perceptual, etc. The focus of this paper is on the issue of noninvertible embedding scheme and on the analysis of two major concerns, the capacity and distortion.

Binning is a coding technique of great significance in information theory. The data classification is reached by means of a parity check matrix in a binning method, referred to as matrix embedding [2], [3], [4]. The binning methods can be roughly classified into informed coding and informed embedding, where the cover object is used as the side information during data embedding. Generally speaking, matrix embedding using a parity check matrix leads to a less level of distortion than those suboptimal embedding algorithm [5], [6], [7], due to the structure of a linear code. Moreover, the data embedded based on parity check matrix can be extracted in the receiver simply by a multiplication operation between the parity check matrix and received vector. Furthermore, with an existing parity check code, the embedding rate with respect to a minimum level of distortion can be determined accordingly. The matrix embedding method is related to the construction of the covering codes with the following two concerns.

1) Finding a good covering codes is equivalent to a good matrix embedding method.
2) A low complexity matrix embedding method in data hiding is equivalent to a fast decoding algorithm of this covering code.

Usually, a maximum likelihood decoding has high decoding complexity for a linear code with large length. Data hiding with matrix embedding from a linear code also suffers the same problem using binning methods. Besides, it is unlikely to employ the optimum embedding, i.e., maximum likelihood algorithm, to find a codeword out of a parity check code. This paper embeds binary data in a manner of sub-optimal embedding that low embedding complexity is attained.

The rest of this paper is organized as follows. In Section II, we briefly discuss the theory and distortion limit of binary data hiding. Section III describes our major work on proposed sub-optimal iterative embedding algorithm. In Section IV, we provide experimental results and constructive discussions. Finally, we state our conclusions in Section V.

II. Bounds on coding theory

The binary data hiding refers to an issue where the average level of distortion d of an embedding strategy can be determined by a binary linear embedding code (n,k) at a given embedding rate $R_e = (n-k)/n$. The lower bound $d_{\min}$ of d is thus estimated using the rate-distortion function of a binary symmetric source. The coding theory related knowledge is discussed here.

A. Rate-distortion function of binary symmetric source

A (n,k) binary linear embedding code C is characterized by the use of a parity check matrix $H \in \{0,1\}^{m\times n}$, where $m = n-k$. Assuming that the code rate is $R = k/n$, the code C is of size $|C| = 2^{nR}$. Given a binary symmetric source (BSS) and a n bit source sequence $u \in \{0,1\}^n$, the Hamming weight distortion is defined as

$$d(\hat{u},u) = \frac{E[d(\hat{u},u)]}{n} = \frac{D}{n}$$

[1]This work was supported by Grant NSC-99-2221-E-005-081-MY2.

where $\hat{u}$ represents a quantized codeword existing in the code C , and $D = E[d(\hat{u}, u)]$ is the average hamming distortion between $\hat{u}$ and u per each n bits simple block. For a good (n, k) linear block code, and sufficiently large n, the minimum average distortion is less than

$$d(\hat{u}, u) \gtrsim \delta$$

and

$$\delta = h^{-1}(m/n) = h^{-1}(1 - R)$$

where $h^{-1}(*)$ is the inverse function of the binary entropy function h. Assuming that D/n represents the average distortion of each bit among a n-bit sequence, D is thus the average distortion of each binary sequence and can be expressed as $D = E[w(e)]$. The lower bound δ of each bit average distortion in blocks can be written as $d = D/n \geq \delta$. When performing the binary data embedding of a sequence of n bits, the embedding efficiency is defined as

$$\eta = \frac{1 - R}{d} = \frac{m}{D}$$

In the case of a matrix embedding use the suboptimal decoding strategy instead of maximum likelihood decoding, the embedding scheme may lose some embedding efficiency due to the suboptimal decoding strategy. For a (n, k) linear code C, the embedding efficiency between both the optimal, i.e. Maximum likelihood decoding, and the suboptimal algorithms can be hence related as

$$\frac{m}{nh^{-1}(R_e)} \geq \frac{m}{D_{opt}} \geq \frac{m}{D_{sub}}$$

where D_{opt} and D_{sub} represent the average distortion estimated for each block in the optimal algorithm and the suboptimal algorithm, respectively. The above equation can be express in an alternative form as $\eta_\delta \geq \eta_{opt} \geq \eta_{sub}$.

B. Optimal embedding algorithm

A code is referred to as syndrome codes due to the use of a parity check matrix. It is built with two main goals, that is, to 1). find a well defined coding structure or a well behaved parity check matrix, and 2). perform decoding through efficient decoding algorithm. Given a host vector and a logo vector intended for embedding, the syndrome of the host vector must be firstly found and then added to that of the logo vector as a way to acquire a toggle syndrome. Ultimately, the coset leader, corresponding to the toggle syndrome, can be found using the ML decoding. The coset leader is then added to the host vector as a way to yield a closest vector, into which a secret logo vector is embedded. This is illustrated as Fig. 1.

A (n, k) linear block code C can be characterized with a parity check matrix H of size $(n - k) \times n$ as follows.

$$C = \{r | Hr = 0\}$$

where the sequence $r \in F_2^n$. Derived from the above equation, the syndrome s of the sequence r, in the case of a nonzero Hr, is defined as $s = Hr$. Furthermore, the set composed of all

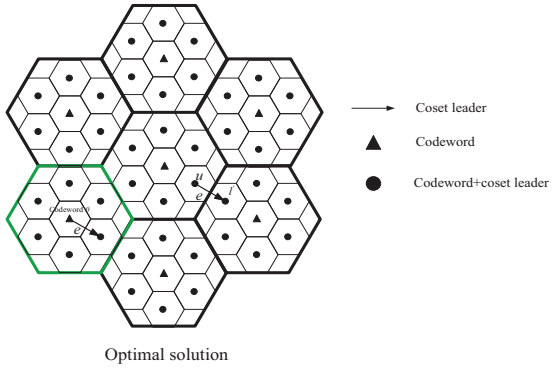

Fig. 1. Geometric interpretation of optimal information embedding.

the sequences r, corresponding to the identical s, is referred to as the coset of the code C, defined as

$$C^s = \{r | Hr = s\} = \{c + e | c \in C\}$$

where e denotes the coset leader in the standard array. s can be derived through H from an arbitrary sequence r, and e can be expressed, in terms of a ML decoding function, as

$$e = f(Hr) = f(s)$$

where $f(\cdot)$ represents the linear codes decoding function. Determined through ML decoding, the coset leader e is added to r as a means to recover the code C, that is closest to the sequence r. There exists a host vector corresponding to an arbitrary sequence u of length n bits within the coset C^u of the standard array. The syndrome $s_u = Hu^T$ corresponding to C^u is referred to as the host vector syndrome. Referred to as the logo vector, a known binary sequence s_l of length $n-k$ bits is intended for embedding. It is known with ease that the coset leader e_{opt} must be located within a set C^x ahead before a sequence, closest to u, with syndrome s_l, is discovered. Then the syndrome s_x is determined by the addition of logo vector s_l to s_u. From the view point of decoding, the coset leader e_{opt} can be discovered through maximal likelihood (ML) decoding, expressed as

$$e_{opt} = f_{opt}(s_u + s_l) = f_{opt}(s_x)$$

Suppose that a sequence $x \in C^x$ exists, and C^x represents a coset of the code C. It is intended to seek x with the minimal weight, that is, $x = e_{opt}$, which is expressed as

$$e_{opt} = \arg \min_{x \in c^x} w_H(x)$$

Once discovered, the coset leader e_{opt} is added to the host vector u as $l' = u + e_{opt}$. Essentially, l' is the sequence, closest to the sequence u within F_2^n dimensional space, and contains the logo vector s_l . Illustrated below is a way to embed a binary linear code through ML decoding by means of a standard array.

Although once e_{opt} is known, the optimal embedding vector l' can be thus discovered. It remains a difficult subject to find

e_{opt} in the case of a long (n, k) linear code C or a large value of k, due to the fact that the complexity of the ML decoding increases as 2^k. A suboptimal embedding algorithm is then proposed here in the next section to replace the ML algorithm toward resolving the above mentioned disadvantage.

III. Sub-optimal embedding algorithm

As a matter of fact, the binary data embedding had been implemented using the suboptimal embedding algorithm in a number of research activities. Proposed in 2002 by Oscar, three types of binary data embedding are DHST, DHPT and DHSPT, in the last two of which data embedding bits are altered in pairs as a way to improve the MPSNR of image quality. These three algorithms can be combined with other binary embedding algorithms. Proposed also by Oscar, tree based parity check (TBPC) algorithm [5], [6], a type of suboptimal embedding algorithm, employs a parity check matrix. [5], [6] embedding algorithm can also been characterized with the sub-optimal embedding algorithm presented here. An illustration, not requiring a parity check matrix, is presented as follows. Proposed by [7], the binary embedding algorithm is of an embedding capacity dependant on the partitioned host vector of size $m \times n$, into which $\log_2(mn+1)$ number of 1's can be embedded, and a maximum of 2 bits can be altered. However, those binary embedding algorithm is just a special case of linear embedding codes, for the reason that an arbitrary parity check matrix is capable of the binary data embedding. Yet, the parity check matrix is designed and the coset leader vector is sought in a measure that affects the embedding distortion. However, those binary embedding algorithm is just a special case of linear embedding codes, for the reason that an arbitrary parity check matrix is capable of the binary data embedding. Yet, the parity check matrix is designed and the coset leader vector is sought in a measure that affects the embedding distortion.

An efficient algorithm is presented here as a means to perform the binary data embedding. Here, the coset leader is found in an alternative way to the conventional ML decoding algorithm. As follows, a simple way is utilized to locate a low weighted toggle vector during the search of coset leaders vector e_{opt}. It is intended to locate a vector e_{sub}, and $w(e_{sub}) \geq w(e_{opt})$, in lieu of the optimal coset leader $w(e_{opt})$. It is requested that e_{sub} stay as close to $w(e_{opt})$ as possible. It is for sure that e_{sub} is a vector defined in C^x. Obtained by the addition of e_{sub} to the host vector u, the target vector l' cannot be assured as the optimal vector. As illustrated in Fig. 2, this is the one referred to as the suboptimal embedding algorithm.

Here, a suboptimal algorithm, i.e. weight approximation embedding (WAE) algorithm, is stated. Unlike the ML embedding algorithm, a minimum weighted toggle vector is searched in an iterative technique. More specifically, in ML algorithm, the minimum weight is gained by an entire search of the codewords within the linear embedding code, while instead WAE algorithm seeks the less weighted toggle vector by times of iterations. Performing k times of hardware operations each

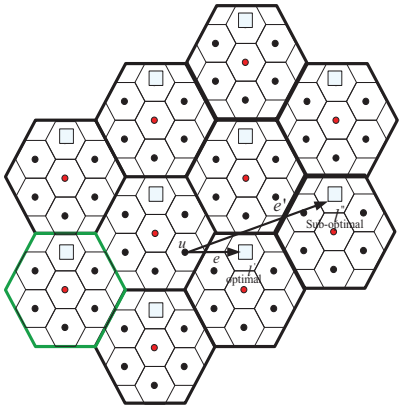

Fig. 2. Geometric interpretation of sub-optimal information embedding.

time, WAE proceeds in an iterative way until the convergence is reached. It provides a lower operation complexity than the ML algorithm does at the cost of distortion efficiency.

Recalling from section II that in the ML algorithm, $f(s_x)$ is decoded to gain the codeword c, and the coset leader vector e is obtained by addition of c to x, i.e. $x + c$. Here, $f(s_x)$ is not directly decoded through the ML decoding, but instead it is intended that the syndrome s_x of the toggle x remains invariant. As a simplest way to achieve this, the codeword c, out of the linear code C, is added to the toggle x as x', i.e. $x' = x + c$. As a result, the weight of x' is altered through the codeword c, but x' still falls within the coset C^x. Although there is a total of number of 2^k codewords c's in the linear code C, it is unrealistic to test them all. It is just that only k number of codewords are selected from among 2^k codewords for test, which form the row vectors g_i of a systematic generator matrix G_s as follows.

$$G_s = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix}$$

These row vectors set is defined as $\Gamma = \{g_1, g_2, \cdots, g_k\}$. It is known that for an arbitrary $g_i \in \Gamma$ and the toggle syndrome s_x is expressed as

$$\begin{aligned} s_x &= H(h+l) \\ &= Hx \\ &= Hx + Hg_i \\ &= H(g_i + x) \\ &= Hx' \end{aligned}$$

where x, h and l is toggle vector, host vector and logo vector, respectively. Although the syndrome s_x remains invariant with g_i added to x, the toggle vector x together with the corresponding weight does change. The distortion will be reduced in the event that a less weighted modified toggle vector x' than the original toggle vector x can be found.

Eventually, the distortion can be improved by means of a small amount of weight variation.

Definitely, the dimension of a candidate Γ can be extended, and k number of row vectors can be selected out of G_s, or $\Gamma = \{g_i | i = 1 \cdots \mathrm{C}_i^k, i \leq 2\}$ can be formed as a combination of two arbitrary vectors within G_s. Yet, the price paid for an increment in i is a higher operation complexity. The case for $i = 1$ is addressed in this work.

Now modified toggle vector $x' = x + g_i$ is gained through an appropriate weight variation of toggle x with the main goal of approaching the weight of x' to that of the coset leader e. Assuming $w_H(x) = \lambda$ and λ is a constant, the vector x' , approaching to e, can be expressed as

$$x' = \arg \min_{g_i \in \Gamma} w_H(g_i + x)$$

where the vector x' represents the vector with the minimum weight after k times of tests. Besides, in case

$$w(e) \leq w_H(x') \leq \lambda$$

then the vector x' stays closer to e than the vector x does. The above mentioned algorithm, designated as WAE, is capable of reducing the toggle weight as much as possible in an iterative way.

IV. SIMULATION RESULTS

As follows, the algorithms stated above are simulated on the embedding efficiency. In experiment, the host image U and logo vector s_l is selected randomly. The host image U is divided into the $n \times N$ non-overlapping blocks, where each block u is $n \times 1$ in size, and each block u embed the logo vector S_l of m bits. According to the experiment conducted above, the WAE algorithm, compared with the ML algorithm, requires a lower operation complexity, with the price of a degraded efficiency $\eta = m/D$. As follows, the efficiency η, corresponding to various systematic linear block codes, is compared between both the WAE and ML algorithms. Applied

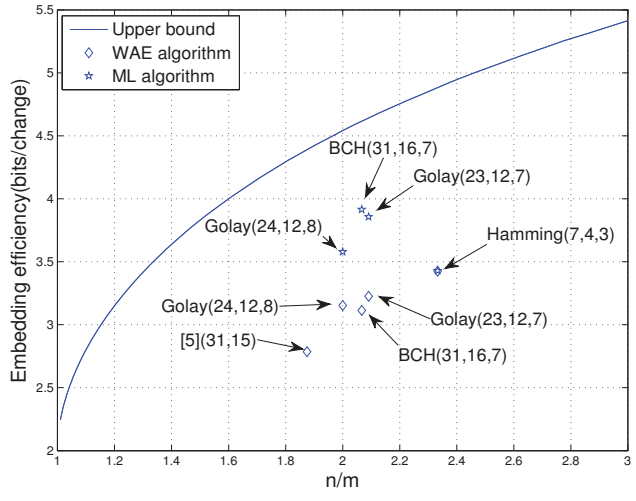

Fig. 3. Embedding effciency of various algorithms.

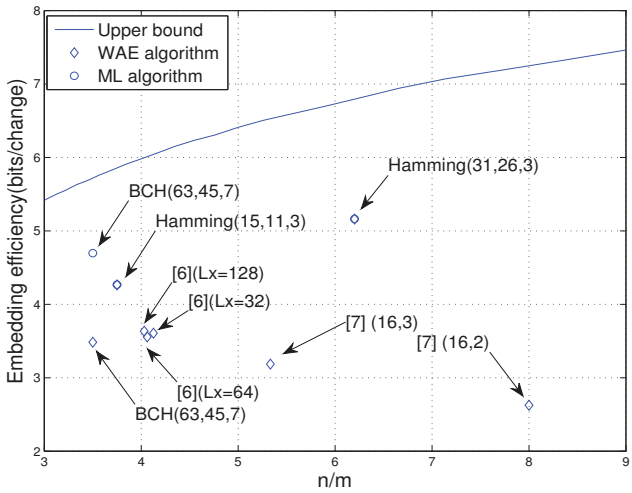

Fig. 4. Embedding effciency of various algorithms.

to a hamming linear code, WAE exhibits an identical efficiency to that by performing the ML algorithm, for the reason that WAE requires merely one time of iteration to successfully locate the toggle coset leader.

V. CONCLUSIONS

Proposed in this work is a suboptimal WAE embedding algorithm, with low operation complexity, made applicable to an arbitrary (n, m) embedding code with a parity check matrix. In most cases, an ML algorithm is criticized for being extremely sensitive to the dimension $(n - m)$, due to the fact that the operation complexity varies exponentially with $(n - m)$. Rather the complexity exhibits a linear dependence on $(n - m)$ when performing WAE, making it applicable to a long linear code embedding. Given a (n, m, λ_{min}) linear embedding code, the complexity required by WAE is merely $O(\overline{\mu}(n - m))$ when locating the coset leader e, whilst it is $O(2^{(n-m)})$ by the ML algorithm, an unacceptable figure for a large value of $(n - m)$. The much lower complexity in WAE is reached merely at the cost of a small deal of embedding efficiency.

REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proc. IEEE*, vol. 87, no. 6, pp. 1062–1078, Jul. 1999.

[2] J. Bierbrauer, On Crandall's Problem [Online]. Available: http://www.ws.binghamton.edu/fridrich/covcodes.pdf 1998.

[3] R. Crandall. Some notes on steganography. *Steganography Mailing List, available from http://os.inf.tu-dresden.de/westfeld/crandall.pdf*, 1998.

[4] F. Galand and G. Kabatiansky. Information hiding by coverings. In *Proceedings ITW2003, Paris, France, 2003*, pages 151–154.

[5] R. Y. M. Li, O. C. Au, K. K. Lai, C. K. Yuk, and S. Y. Lam, "Data hiding with tree based parity check," *in Proc. IEEE Int. Conf. Multimedia and Expo (ICME 07),* 2007, pp. 635–638.

[6] R. Li, O. Au, C. Yuk, S. Yip, and S. Lam, "Halftone image data hiding with block-overlapping parity check," *in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP),* 2007, vol. 2, pp. 193–196.

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