Enhancing and Constructing Two Way Matrix Embedding For Efficient Embedding

[1] *P. Subhadra,* [2] *A. Sai Kumar*

[1]Associate Professor in CSE Dept, VCE,Hyderabad.
subhadra.perumalla@gmail.com

[2]M.Tech (C.S.E), VCE,Hyderabad.
a.saikumar1989@gmail.com

Abstract

In designing steganographic schemes, matrix embedding is an efficient method for increasing the embedding efficiency that is de- fined as an average number of bits embedded via per change on the cover. Matrix embedding is a previously introduced coding method that is used in steganography to improve the embedding efficiency (increase the number of bits embedded per embedding change). Higher embedding efficiency translates into better steganographic security. This gain is more important for long messages than for shorter ones because longer messages are in general easier to detect. In this paper, we present a new approach called two way embedding Compared with the original matrix embedding, the proposed method can exponentially reduce the computational complexity for equal increment of embedding efficiency. Experimental results also show that this novel method achieves higher embedding efficiency and faster embedding speed than previous fast matrix embedding methods, and thus is more suitable for real-time steganogaphic systems.

Index Terms— Two way Embedding efficiency, covering codes, embedding speed, matrix embedding, steganography.

1. Introduction

The main requirement for a steganographic scheme is statistical undetectability. Given the knowledge of the two way embedding algorithm and the source of cover media, the attacker should not be able to distinguish between stego and cover objects with success rate better than random guessing. Steganographic security is mostly influenced by the type of cover media, the method for selection of places within the cover that might be modified, the type of embedding operation, and the number of embedding changes, which is a quantity closely related to the length of the embedded data.

Given two embedding schemes that share the first three attributes, the scheme that introduces fewer embedding changes will be less detectable**.** Matrix embedding is a general principle that can be applied to most steganographic schemes to improve their embedding efficiency, which is defined as the expected number of random message bits embedded per one embedding change Fridrich *et al* proposed random linear codes with small dimensions, which can achieve high embedding efficiency for only large embedding rates. However, codes for large embedding rates are important because the construction imp lies that small embedding rate codes can be generated from large embedding rate codes. What is more, the large It was made popular by Westfeld who incorporated a specific implementation of matrix embedding using binary Hamming codes in his algorithm. It is intuitively clear that the gain in embedding efficiency can be larger for short messages than for longer ones. Since in general short messages are more difficult to detect than longer ones.

Here In the present paper, we propose a two way embedding in this terminology and basic concepts of linear codes necessary to explain the embedding methods.We refer to this new method as matrix extending because we design the fast algorithm by appending some referential columns to the parity check matrix. Analysis and experimental results show that the proposed method can flexibly trade embedding efficiency for embedding speed, or vice versa. Compared with the original matrix embedding the proposed method can exponentially decrease computational complexity by increasing the number of the referential columns while achieving an equal increment of embedding efficiency.Compared with the fast matrix embedding methods the novel method can reach higher embedding efficiency with faster embedding speed.

II. TWO WAY MATRIX EMBEDDING:

The data embedding codes usually are measured by embedding efficiency versus embedding rate. Embedding rate is defined as the average number of bits embedded into each pixel, and embedding efficiency is defined as the average number of bits embedded by per embedding change. that is why the embedding efficiency can be improved when increases, but the computational complexity of searching for this solution exponentially grows. In this section, we propose two embedding matrix so exponentially expand the solution space, but only cost linearly increasing time to search the solution space. The key idea of the proposed method is to append some referential columns to the matrix Multiplication by recursively partitioning into smaller blocks.

A11 A12
A21 A22
B11 B12
B21 B22

requires only multiplications:

M1 := (A11 + A22)(B11 + B22)
M2 := (A21 + A22)B11
M3 := A11(B12 B22)
M4 := A22(B21 B11)
M5 := (A11 + A12)B22
M6 := (A21 A11)(B11 + B12)
M7 := (A12 A22)(B21 + B22):

On one hand, the solution space of is twice as large as that On theother hand, from each solution we can construct two corresponding solutions of by appending a “0” or appending a“ 1 ” and flipping the first bits as shown in . Thus we can find the minimal weight solution of by only exhausting the solution space of in the following manner.

Theorem 1: (two way Matrix embedding) Let C be an [n, k] code with a parity check matrix H and covering radius R. The embedding scheme below can communicate n−k bits m ∈ F n 2 in n pixels with bits x using at most R changes:

$$\boldsymbol{Emb(x,m)=x+eL(m \,\square\, Hx)=y,}$$

$$\boldsymbol{Ext(y)=Hy,}$$

$$\boldsymbol{\text{where } m \,\square\, F}$$

$$\boldsymbol{n \square k\ q}$$

is a sequence of n − k message symbols and eL(m−Hx) is a coset leader of the coset C(m−Hx). Indeed, since C has covering radius R, we know that d(x, y) = w(eL(m−Hx)) ≤ R, which shows that the embedding scheme has (a tight) distortion bound R. To see that Ext(Emb(x,m))=m, note that Ext(Emb(x,m))=Hy=Hx + HeL(m−Hx) = Hx + m−Hx = m

For an embedding scheme realized using matrix embedding, the expected number of embedding changes for messages uniformly distributed in F n−k 2 is equal to the average weight of all coset leaders of C. It is reasonable to assume that the messages are drawn uniformly at random from F n−k 2 since typically they will be encrypted before embedding Because any two words x, y from the same coset C have the same distance from C

Algorithm: Embedding M bits in an N-element cover object using random linear codes.

1) To embed M bits in an N-element cover object, first find n such that αn≥M/N>αn-1,

2) Read the next n−k bits x from the cover object (along a stego key dependent path) and the next message segment m of the same length.

3) Find any e that solves He = m −Hx.

4) In the list of all 2k codewords, find the closest code word to e, denote c(e).

5) [Embedding modifications]

y =x+e−c(e) is the stego object.

6) If we are at the end of the cover object, stop, otherwise go to 1.

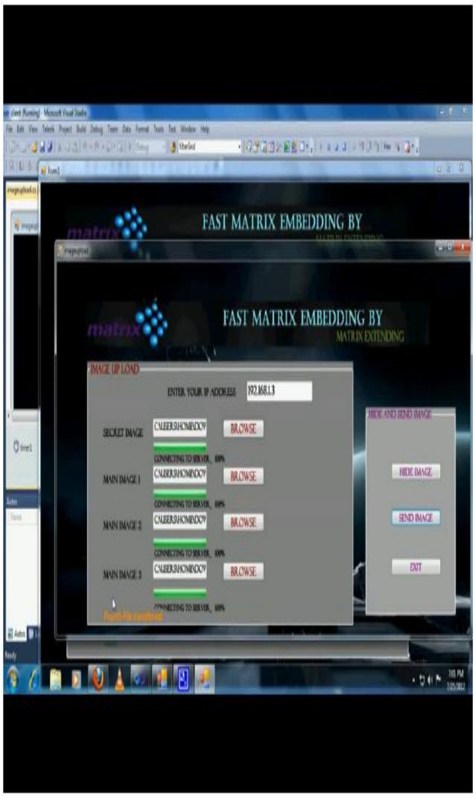
7) [Extraction step] The message bits are extracted by following the same embedding path and calculating n−k bits m from each block y of the stego object

m = Hy.

A. Comparison With Original Matrix Embedding

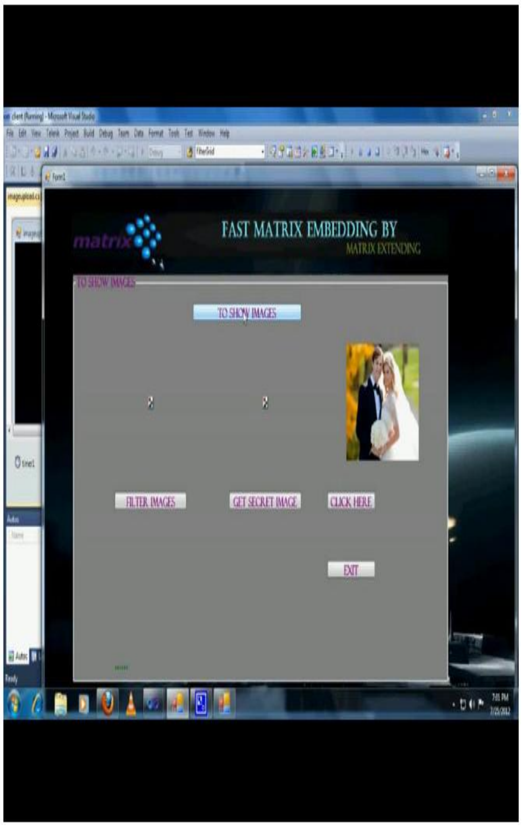
In This paper We give a small example of how fast the embedding based on random codes runs on a computer. We simulated embedding into an image with N=1280×1024 pixels using a random code with block length n = 100. We measured the time taken to perform the embedding with dimensions k = 10, 12, and 14. The test was performed on Pentium IV running at 3.4 GHz with 1 GB RAM. The algorithm was implemented in C++ and compiled under Linux with GCC. In Algorithm to find an optimal solution in the expanded solution space with a size of two matrix mathematical representation we only need to search a small solution space with a size of , and do comparisons and additions by (20) for each solution. Therefore, the computational complexity is equal to , which linearly increases with , and thus Algorithm can achieve equal embedding efficiency with faster embedding speed when compared with the original matrix embedding To verify the conclusion above, we embed messages into a random cover by using the original matrix embedding of and Algorithm, respectively. For the method we take and vary the cover block length from 66 to 280 to get various embedding rates. For Algorithm we take and also vary from 66 to 280. For each embedding rate, we embed 1000 blocks of random messages, and calculate embedding efficiency by the average number of changes.

As shown in Fig. 1,


ENTER YOUR IP ADDRESS 192.168.1.3

FIG(1):UPLOADING THE IMAGES

The two methods achieve equal embedding efficiency [Fig. 1], while the embedding speed of the proposed method significantly outperforms the original matrix embedding [Fig.2]. The embedding speed is measured by Kbits of covers per second for four kinds of embedding rates. The test was performed on Intel Core i5 running at 2.67 GHz with 4-GB RAM. The algorithm was implemented in C and compiled under Microsoft Visual Studio 2008.In the above image we can magnificently find out that going to be upload the images like original image as well as embedding image. upon those two images only we are performs two way embedding algorithm based on pixel values.



FIG(2):DIFFERENT OPERATIONS ON IMAGES

B. Comparison With Fast Matrix Embeddings:

We also compared Algorithm with previous fast matrix embedding and Gao *etal.* proposed a specific parity check matrix, which can embed messages with linear computational complexity according to the length of the cover block. The Majority-vote Parity Check (MPC) method is an improving version of the Tree Based Parity Check (TBPC) method, which embeds messages with linear computational complexity according to the length of the message block. The embedding efficiency of both methods is lower than the original matrix embedding with dimension. To compare with methods in and, we take and in Algorithm As shown in Fig. 2, the proposed method can reach higher embedding efficiency as well as faster embedding speed.

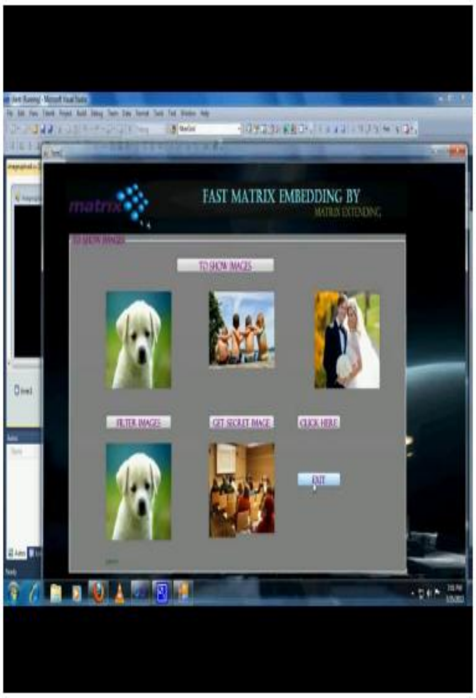


Fig. 3. Comparison of embedding efficiency obtained by two kinds of image.

In the above screen we clearly found out that upper images are image and in lower images one is filtered image and other one is secret image. we are embedding our secret image into 3 main images We take, and embed messages with various embedding rates in two manners by setting uniform segment sizes with and random segment sizes, respectively. As shown in Fig. 3, random segment sizes will decrease the embedding efficiency. On the other hand, from Algorithm, it is obvious that segment sizes have no effect on computational complexity. We review a few relevant known facts about embedding schemes and covering codes that appeared in image screen and establish some more terminology. To see how much the coding improves the embedding efficiency, let us take two relative payloads 0.9 and 0.8. From Figure 1, using random linear codes of dimension 14, Any structured codes with low dimension and fast decoding algorithms that are quantizers can be used for our purpose this should improve the results of embedding of all those images for small code lengths, codes with the smallest average distance to code may not necessarily have the smallest covering radius. We plan to elaborate on this issue in our future work.

CONCLUSION:

In this paper, we present two simple coding schemes suitable for matrix embedding of large amount of data hiding. The codes can be applied to most steganographic schemes without any other changes to their embedding mechanism to increase their embedding efficiency the expected number of random bits embedded using one embedding change. This will improve their steganographic security.

REFERENCES:

[1] M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In **E.Delp** III and **P.W.Wong**, editors, Proceedings IE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents ,an ose, CA, January 16–19, (to appear),feb 2006.

[2] X. Zhang, “Efficient data hiding with plus-minus one or two,” *IEEESignal Process. Lett.*, vol. 17, no. 7, pp. 635–638, Jul. 2010.

[3] M. H. Shirali-Shahreza and S. Shirali-Shahreza, “Real-time and MPEG-1 layer III compression resistant steganography in speech,” *IET Inf. Security*, vol. 4, no. 1, pp. 1–7, 2010.

[4] Y. Gao , X. Li , and **B. Yan** g , “Constructing specific matrix for efficient matrix embedding,” in *Proc. IEEE Int. Conf. Multimedia and Expo*,2009, pp. 1006–1009.

[5] 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

[6] F. J. M. Williams and **N. J. Sloane**. The Theory of Error-Correcting Codes. North-Holland, Amsterdam, 1977.