

Steganography Based on Grayscale Images Using (5, 3) Hamming Code

Cheonshik Kim¹(✉) and Ching-Nung Yang²

¹ Department of Digital Media Engineering, Anyang University,
Anyang-si, Gyeonggi-do, Korea
mipsan@paran.com

² Department of Computer Science and Information Engineering, National Dong
Hwa University, Hualien, Taiwan, ROC
cnyang@mail.ndhu.edu.tw

Abstract. Steganography is a technique to hide secret data in cover images securely. This technique is used for secret communication. However, steganography is not as strong as watermark against various attacks. “Hamming+1” scheme is a well known scheme in the steganography. In this paper, we propose new data hiding scheme that showed better performance compared to “Hamming+1”. The proposed scheme conceals 3 bits per 5 pixels of an image. The experimental result showed that the proposed scheme achieves an 0.599 bpp embedding payload and a higher visual quality of stego images compared to the previous schemes.



Keywords: Steganography · Data hiding · LSB · Hamming code

1 Introduction

Nowadays, due to the advancement of the computer and network technology, people can easily send or receive secret information in various forms to or from almost any places through the Internet. However, important secret messages may leak out while they are being transmitted or exchanged over public communication channel (i.e., e-mail, ftp, web browser) [1, 2]. Therefore, achieving safe secret communication is an important field of research. Recently, many people have started using cryptography and steganography [1–7] to protect their precious data from the attackers. Steganographic technologies are a very important part of future Internet security and privacy on open systems, such as the Internet. The steganography is the least secure means by which to communicate secretly because the sender and receiver can rely only on the presumption that no other parties are aware of the secret message. Using Steganography, information can be hidden in carriers, such as images.

Steganography can be classified into two domains, the spatial [1, 2] and frequency [1, 2]. In the spatial domain, the secret message is inserted directly into the pixels. In the frequency domain, the common method of data hiding are discrete cosine transform (DCT)-based method, discrete wavelet transform (DWT)-based

method, or other methods based on similar mechanisms. The most common methods are histogram-based and least-significant bit (LSB) techniques in the spatial domain. The embedding by flipping LSB of pixels is relatively easy to detect even at very low embedding rates. Essentially, senders and receivers agree on a steganographic system and a shared secret key that determines how a message is encoded in the cover medium. Steganographic schemes have a vulnerable point from steganalysis attacks; thus, many researchers proposed various schemes.

Chang *et al.* [4] proposed (7, 4) Hamming code for data hiding, which improves on the “Hamming+1” scheme. Westfeld’s F5 [5] is the first implementation of matrix encoding. OutGuess [5] was the first attempt to explicitly match the DCT histogram. Crandall [9] showed that linear codes could markedly improve the embedding efficiency. Rongyue *et al.* [14] proposed an efficient BCH coding for steganography, which embeds the secret information inside a block of cover data. The CPT scheme [15] showed the embedding efficiency by hiding messages based on the weighted value of a block. Zhang *et al.* [6] proposed the ternary Hamming codes using the concept of efficiency by exploiting the modification direction (EMD) [13, 16, 17]. The performance of “+/- steganography” was introduced by the [8]. Mielikainen [8] presented a method based on a pair of two consecutive secret bits [11].

In this paper, we proposed (5, 3) hamming code scheme, which can be used to embed 3 bits per 5 pixel blocks in an image. Our proposed scheme is very efficient and steganographic, so it can be used in many various application fields. The rest of this paper is organized as follows. In Sect. 2, we review related previous research schemes. In Sect. 3, we introduce our proposed scheme for grayscale images. In Sect. 4, we explain the experimental results. Section 5 presents our conclusions.

2 Related Works

2.1 Error Correction Code

Many secret data are sent through noisy channel, and it is common for an occasional bit to flip. The channel is “noisy” in the sense that what is received is not always the same as what was sent. Thus if binary data is being transmitted over the channel, when a 0 is sent, it is hopefully received as a 0 but sometimes will be received as a 1 (or as unrecognizable). Noise in deep space communications can be caused, for example, by thermal disturbance. If no modification is made to the message and it is transmitted directly over the channel, any noise would distort the message so that it is not recoverable. The basic idea is to add some redundancy to the message in hopes that the received message is the original message that was sent. The redundancy is added by the encoder and the redundancy message is called a codeword c .

2.2 Hamming Code

Linear codes with length n and dimension k will be described as $[n, k]$ codes. Hamming codes are linear codes and are described as a $[n, k]$ q -ary Hamming

code, where q is the size of the base field, F_q . A generator matrix G for an $[n, k]$ linear code c (over any field F_q) is a k -by- n matrix for which the row space is the given code. In other words, $c = \{xG|x\}$. Matrix encoding conceals messages with the parity check matrix of a linear codes. If H is the checker matrix for c , H is an $(n-k)k$ matrix, the rows of which are orthogonal to c and $\{x|Hx^T = 0\} = c$. This matrix contains a canonical generator matrix G and a parity-check matrix H . For (7,4) code, the matrices are as follows [10].

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad (1)$$

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (2)$$

The 4-bit information word is the first bit-wise multiplied (logically anded) by each column in the generator matrix G . Each bit in each of the column products is then added, modulo-2, to create a parity bit for each product. An example of a 4-bit infoword $p = [1010]$ and its generated codeword is $c = [0011010]$.

$$c = Gp^T \quad (3)$$

In Eq. (3), c is the codeword, which includes redundancy for error checker.

$$s = Hc^T \quad (4)$$

In Eq. (4), the vector Hc^T is called the syndrome (or error syndrome) of the vector c . If the syndrome is zero, then c is a codeword; otherwise, the syndrome represents one of the integers $1, 2, \dots, 7$ in binary. This tells us which of the seven bits of c to switch to recover a valid Hamming codeword from c .

[Example 1] For information sequence $q = [0110]$, we get the following transmitted codeword of length 7. Now all we have to do is multiply the information vector q with matrix G to get a codeword c .

[Example 2] We assume that the codeword c is $[1101001]$. It is easy to calculate the syndrome using Eq. (4) with the parity check matrix H and the codeword $= ([000])^T$. If the syndrome is zero, then c is a valid code; otherwise, c is a wrong code so we need to switch to recover a valid Hamming codeword from c . In this example, s is zero, so there is no need to flip any bit.

3 Our Proposed Scheme

3.1 (5, 3) Hamming Code

Every codeword in a Hamming code [5,3] has a distance of minimum 3 to another codeword, making it possible to correct 1 corrupt bit in each codeword. Let us assume that the sender is transmitting data r to the receiver. The r consists of codeword c and error bit e , as shown in Eq. (5).

$$r = c + e \quad \text{[Image: yellow speech bubble icon]} \quad (5)$$

For example, if the codeword is $r = [1 \ 0 \ 0 \ 0 \ 0]$, the matrix in Eq. (6) (error detection matrix) and Eq. (7) can be used to identify an error in it. In this case, the first bit is the error. If the error is removed, it becomes $r = c$. That is, Eq. (6) is a matrix that is used to identify errors in [5,3] Hamming code.

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Equation (7) is used to verify syndrome.

$$s(r) = rH^T = (c + e)H^T = eH^T \quad \text{[Image: yellow speech bubble icon]} \quad (7)$$

In Eq. (7), rH^T is called syndrome and relies on error vector.

Table 1. Represents a relationship between r and the syndrome table

Order	Error vector	Syndrome
0	[0 0 0 0 0]	[0 0 0]
1	[0 0 0 0 1]	[0 0 1]
2	[0 0 0 1 0]	[0 1 0]
3	[0 0 1 0 0]	[1 0 0]
4	[0 1 0 0 0]	[0 1 1]
5	[1 0 0 0 0]	[1 0 1]
6	[1 1 0 0 0]	[1 1 0]
7	[0 1 1 0 0]	[1 1 1]

$s(r)$ means a syndrome of r . Therefore, the error correction algorithm is as follows.

- The syndrome is computed by Eq. (7) with the codeword of receiver.
- The errors $s(r)$ are searched in Table 1.
- Flipping error position in codeword r .

Suppose that a codeword c is sent and the received vector is $r = c + e$ (addition modulo 2). The first decoding step is to compute the syndrome $s = rH^T = (c + e)H^T = eH^T$. The error position i is the column i of H that is equal to the (transpose of) the syndrome s^T .

Algorithm 1. Hamming Code Decoding

- Step 1: For the received codeword r , compute the syndrome $s = rH^T$.
 Step 2: If $s = 0$, then the decoded codeword is $r = c$.
 Step 3: If $s \neq 0$, then let i denote the column of H , which is equal to s^T .
 Step 4: There is an error in position i of r . The decoded codeword is $r = c + e_i$, where e_i is a vector with all zeros, except for a 1 in the i th position.

This decoding procedure fails if more than one error occurs.

3.2 Encoding Procedure

Our scheme is described below in terms of the embedding procedure for hiding secret data in a grayscale image. A cover image is divided into non-overlapping 5-pixel blocks. We present step-by-step description of the embedding procedure:

Input. Cover image I size $H \times W$, a binary secret message δ of maximum length $H \times W - 1$, and the parity check matrix H , $\text{cnt} = \lfloor (H \times W)/5 \rfloor$.

Output. Stego image I' size $H \times W$.

Step 1: Divide cover image I into 1×5 blocks, let $r_i = (b(x_1), \dots, b(x_n))$, where $b(\cdot)$ denotes LSB of a pixel. Further, c denotes codeword and a set of LSB.

Step 2: Read partitioned pixels and secret messages into array variable x and δ , respectively.

Step 3: Calculate the syndrome s by applying Eq. (7) to the parity check matrix H and c , i.e., $s = rH^T$. Compute $s = s \oplus \delta_j^k$, where \oplus is XOR operation and $j = 1 \dots n, j = j + k$. The pos is the position for error correction.

Step 4: Find the syndrome corresponding to pos in Table 1. Then, find the row of pos in Table 1 and flip the value in that position in the corresponding error vector in r . That is, $0 \leftarrow 1$ or $1 \rightarrow 0$.

Step 5: $\text{cnt} = \text{cnt} - 1$; If cnt is not 0, go to Step 2.

Step 6: Return the completed stego image.

3.3 Decoding Procedure

Our scheme is described below in terms of the extracting procedure of secret message bits from the stego image. A stego image is divided into non-overlapping 5-pixel blocks. We present step-by-step description of the extracting procedure:

Input. Stego image I' sized $H \times W$ and parity-check matrix H , $\text{cnt} = \lfloor (H \times W)/5 \rfloor$.

Output. A secret message δ .

Step 1: Divide cover image I into 1×5 blocks, let $r_i = (b(x_1), \dots, b(x_n))$, where $b(\cdot)$ denote LSB of a pixel. Let c denote codeword and a set of LSB.

Step 2: Read partitioned pixels and secret messages into array variable x and δ , respectively.

Step 3: Calculate the syndrome s by applying Eq. (7) to the parity check matrix H and c , i.e., $s = rH^T$. The calculation result is the hidden message bit, and its value is assigned to δ in the following way.

$$\delta_i = \delta_i + rH^T$$

Step 4: $cnt = cnt - 1$; If cnt is not 0, return to Step 2.

Step 5: Return when operation is complete.

[Example 3] We assume that the codeword is $r = [11001]$ and secret code is $\delta = [101]$. Apply the Eq. (7) to the codeword and secret code, $pos = exor(rH^T, \delta)$. In this case, $pos = 2$. Find the syndrome pos in Table 1 and search the error vector in the corresponding row of pos . Since pos is 2, the position of '1' in the corresponding error vector is fourth from the left. Therefore, flip the 4th value in r to complete the encoding procedure. That is, $r = [11011]$. From the receiver, it is easy to find the concealed secret bits by simply solving Eq. (7) for r .

4 Experimental Results

We proposed a (5,3) Hamming scheme for data hiding. To prove our proposed scheme is correct, we performed an experiment to verify that it ensures the restoration of the hidden image. In addition, the quality of stego image is very important for resisting detection from attackers. Therefore, our method is feasible for making good quality stego images from the original grayscale image. To



carry out our experiment, 512512 grayscale images were used as cover images. Figure 1 shows cover images [12] for experiment to verify our proposed scheme.

The two most important elements in the experiment are quality of stego images (resolution, PSNR) and the embedding rate of the data stored in the stego image. Such evaluation criteria are commonly used to evaluate the performance of data embedding. In this paper, PSNR (Peak Signal to Noise Ratio) [13] was used to prove such evaluation more objectively.

$$PSNR = 10 \log_{10} \left(\frac{I_{max}^2}{MSE} \right), \quad (8)$$

That is, in Eq. (8), the Eq. (9) was applied to the value of the difference between the original image I and stego image I' .

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (|I_{i,j} - I'_{i,j}|)^2, \quad (9)$$

Here, M is the width of the image and N is the height. High PSNR means the stego image is similar to the original image and low PSNR means no resemblance



Fig. 1. The original images used in the experiment

between the original and stego images. In general, PSNR over 30 dB, indicates that it is not easy to detect stego images visually. In order to measure the capacity of data hiding in stego images, *bpp* (bits-per-pixel) means embedding payload. Embedding rate is calculated using the Eq. (10).

$$p = \frac{\|S\|}{M \times N} (bpp), \quad (10)$$


In Eq. (10), p denotes bpp , which is an embedding payload. Our experiment compares the number of secret bits that can be carried by a cover pixel. $|d|$ is the number of bits of a secret message d .

Table 2. Comparison between the performance of “Hamming+1” scheme and the proposed scheme

Method Images	Hamming + 1		Our proposed scheme	
	PSNR	p	PSNR	p
Baboon	53.71	0.499	54.61	0.599
Barbara	48.6	0.499	54.63	0.599
Boats	49.37	0.499	54.62	0.599
Goldhill	53.73	0.499	54.6	0.599
Jet (F16)	51.61	0.499	54.62	0.599
Lena	52.43	0.499	54.62	0.599
Pepper	47.26	0.499	54.63	0.599
Tiffany	47.46	0.499	54.64	0.599
Zelda	54.04	0.499	54.6	0.599
Average	50.91	0.499	54.6	0.599

Figure 2 shows the stego images from the results of the experiments and their respective PSNR. Table 2 shows the comparison between the performance of “Hamming+1” scheme and the scheme proposed in this paper. “Hamming+1” scheme has the average data embedding rate of 0.499 and the average PSNR for the cover image of 50.91 dB. However, the scheme proposed in this paper showed better performance in the experiment, as the data embedding rate was 0.599 and the average PSNR was 54.62 dB. This proves that our proposed scheme is better compared to “Hamming+1”.

Table 3. Connection between change density and embedding rate

	k	n	Change density	Embedding rate	Embedding efficiency
	1	1	50.00 %	100.00 %	2
	2	3	25.00 %	66.67 %	2.67
	3	5	16.67 %	60.00 %	3.60
	3	7	12.50 %	42.86 %	3.40
	4	15	06.25 %	26.67 %	4.27
	5	31	03.13 %	16.13 %	5.16
	6	63	01.56 %	9.52 %	6.10
	7	127	00.78 %	5.51 %	7.06

The embedding efficiency of the $(1, n, k)$ code is always larger than k . Table 3 shows that the rate decreases with increasing efficiency [5]. Hence, we can achieve

high efficiency with very short messages. Our proposed (5, 3) code scheme show higher embedding rate and efficiency compared to (7, 3) hamming code.



Fig. 2. The original images used in the experiment

Figure 3 shows the result from the steganalysis [18] with a database of 200 grayscale images. The circle is the original grayscale image, and crosses are stego images. The crosses are located around the circles, so it is impossible to detect our proposed method from this steganalysis tool. Therefore, our method is very strong related security.

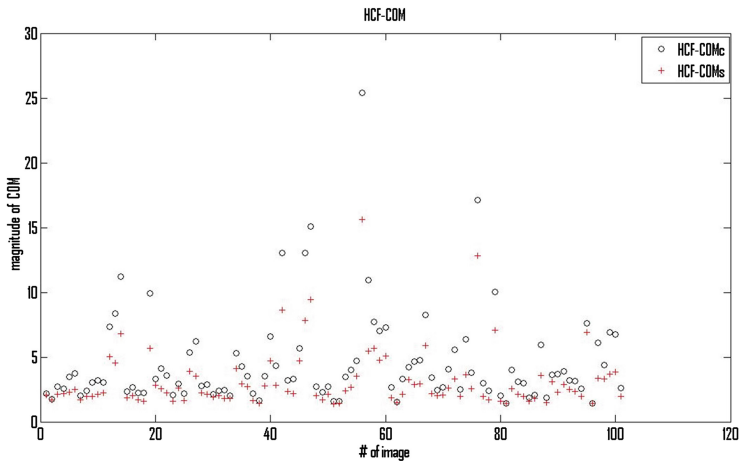


Fig. 3. Cover image (circles) and Stego image (crosses) after embedding for 200 images

5 Conclusions

Matrix Coding and “Hamming+1” are steganographic data hiding schemes. Therefore, these schemes are very good safety channel in many dangerous communication environment. Our scheme is motivated from the matrix coding method. This paper described the use of steganography and its functions. The paper also introduced researches related to this topic and provided explanations for error correction code and Hamming codes. Finally, we proposed a steganographic data-hiding scheme. We proved that our proposed scheme performs better (embedding rate and image quality) compared to “Hamming+1” scheme.

Acknowledgement. This research was supported by the Basic Science Research Program Through the National Research Foundation of Korea (NRF) by the Ministry of Education, Science and Technology (20120192).

References

1. Bender, W., Gruhl, D., Mormoto, N., Lu, A.: Techniques for data hiding. *IBM Syst. J.* **35**, 313–336 (1996)
2. Provos, N., Honeyman, P.: Hide and seek: an introduction to steganography. *IEEE Secur. Priv.* **1**(3), 32–44 (2003)
3. Zhang, W., Wang, S., Zhang, X.: Improve embedding efficiency of covering codes for applications in steganography. *IEEE Commun. Lett.* **11**(8), 680–682 (2007)
4. Chang, C.-C., Kieu, T.D., Chou, Y.-C.: A high payload steganographic scheme based on (7, 4) Hamming code for digital images. In: *International Symposium on Electronic Commerce and Security*, Guangzhou, China, pp. 16–21 (2002)

5. Westfeld, A.: F5: A steganographic algorithm. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)
6. Zhang, X., Wang, S.: Efficient steganographic embedding by exploiting modification direction. *IEEE Commun. Lett.* **10**(11), 781–783 (2006)
7. Chang, C.C., Chen, T.S., Chung, L.Z.: A steganographic method based upon JPEG and quantization table modification. *Inf. Sci.-Informatics Comput. Sci.* **141**(1–2), 123–138 (2002)
8. Mielikainen, J.: LSB matching revisited. *IEEE Signal Process. Lett.* **13**(5), 285–287 (2006)
9. Crandall, R.: Some notes on steganography. Steganography Mailing List (1998). <http://os.inf.tu-dresden.de/westfeld/crandall.pdf>
10. Hamming, R.W.: Error detecting and error correcting codes. *Bell Syst. Tech. J.* **29**(2), 147–160 (1950)
11. Chan, C.S., Chang, C.Y.: Hiding secret in parity check bits by applying XOR Function. In: 2010 9th IEEE International Conference on Cognitive Informatics (ICCI), pp. 835–839 (2010)
12. University of Southern California. The USC-SIPI Image Database. <http://sipi.usc.edu/database/>. Accessed 1 March 2011
13. Kim, C.: Data hiding by an improved exploiting modification direction. *Multimedia Tools Appl.* **69**(3), 569–584 (2014)
14. Zhang, R., Sachnev, V., Botnan, M.B., Kim, H.J., Heo, J.: An efficient embedder for BCH coding for steganography. *IEEE Trans. Inf. Theory* **58**, 7272–7279 (2012)
15. Huy, P.T., Kim, C.: Binary image data hiding using matrix encoding technique in sensors. *Int. J. Distrib. Sens. Netw.* **2013**, 1–7 (2013). Article ID. 340963
16. Kim, H.J., Kim, C., Choi, Y., Wang, S., Zhang, X.: Improved modification direction methods. *Comput. Math. Appl.* **60**(2), 319–325 (2010)
17. Kim, C., Shin, D., Shin, D., Zhang, X.: Improved steganographic embedding exploiting modification direction in multimedia communications. In: Park, J.J., Lopez, J., Yeo, S.-S., Shon, T., Taniar, D. (eds.) STA 2011. CCIS, vol. 186, pp. 130–138. Springer, Heidelberg (2011)
18. Goljan, M., Soukal, D.: Higher-order statistical steganalysis of palette images. In: Proceedings of the SPIE, Electronic Imaging, Security, Steganography, Watermarking of Multimedia Contents V, Santa Clara, California, pp. 178–190 (2003)