Chapter 80
Improving Data Hiding Capacity Based on Hamming Code

Cheonshik Kim and Ching-Nung Yang

Abstract Matrix encoding proposed by Crandall can be used in steganography data hiding methods. Hamming codes are kinds of cover codes. “Hamming + 1” proposed by Zhang et al. is an improved version of Matrix encoding steganography. The embedding efficiency of “Hamming + 1” is equal to $(k + 1)2^{k+1} - 1/(2^{k+1} - 1)$, and embedding rate is $(k + 1)/2^k$. Our proposed “Hamming + 3” scheme has a slightly reduced embedding efficiency, but improve highly embedding rate. We therefore propose verifying the embedding rate during the embedding and extracting phase. Experimental results show that the reconstructed secret messages are the same as the original secret messages, and that the proposed scheme exhibits a good embedding rate compared to that of previous schemes.

Keywords Watermark · Steganography · Matrix encoding · Hamming codes

80.1 Introduction

The purpose of data hiding [1, 2] is to facilitate covert communication in the form of concealed messages in a cover media to modify the media. In the case of a single carrier for an application, all secret information such as images, videos, and MP3 files is stored in the carrier. The goal of data hiding is to ensure that embedded data remain inviolate and recoverable. There are two issues with data hiding. One is to provide proof of the copyright, and the other is to provide

C. Kim (✉)
Department of Digital Media Engineering, Anyang University, Anyang, Republic of Korea
e-mail: mipsan@paran.com

C.-N. Yang
Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan
e-mail: cnyang@mail.ndhu.edu.tw

J. J. (Jong Hyuk) Park et al. (eds.), *Frontier and Innovation in Future Computing and Communications*, Lecture Notes in Electrical Engineering 301,
DOI: 10.1007/978-94-017-8798-7_80, © Springer Science+Business Media Dordrecht 2014

assurance of content integrity. Therefore, the data should stay hidden in a host signal, even if that signal is subjected to manipulation or degrading such as filtering, re-sampling, cropping, or lossy data compression. However, data hiding generally shows weakness to such manipulation. There are trade-offs between the quantity of embedded data and the degree of immunity to host signal modification. As one increases, the other must decrease. Although this can be shown mathematically for some data-hiding systems such as a spread spectrum, it seems to hold true for all data-hiding systems. The goal of steganalysis is to detect (and possibly prevent) such communication. Generally, steganalysis tools can easily detect a stego image when the error rates are over about 10 % to conceal a message.

Crandall [3] proposed a new data hiding scheme called matrix encoding. The F5 algorithm [4] is based on matrix encoding and implemented by the Westfeld. We can the definition of the cover coding [5–7] in [4]. Matrix encoding was also used in large payload applications [8]. BCH codes were applied to achieve a tradeoff between embedding complexity and efficiency [9]. Westfeld showed matrix encoding using Hamming codes. The CPT method [10] shows the embedding efficiency by hiding messages based on the weighted value of a block. Matrix encoding and CPT can be applicable to LSB steganography. Zhang and Wang [11] showed the ternary Hamming codes using the concept of efficiency by exploiting the modification direction (EMD). The performance of "±steganography" was introduced by the [12]. Mielikainen [13] presented a method based on a pair of two consecutive secret bits. Chang et al. [14] proposed (7, 4) Hamming code for data hiding, which improves on the "Hamming + 1" scheme.

In this paper, we propose novel improving data hiding methods by extending the Hamming codes. Our proposed method can significantly improve the embedding rate of "Hamming + 1" scheme, and perform equally well, or even outperform.

The rest of this paper is organized as follows. In Sect. 80.2, we review current and related work. In Sect. 80.3, we introduce our proposed "Hamming + 3" for grayscale images. In Sect. 80.4, we explain the experimental results. Section 80.5 presents our conclusions.

80.2 Related Works

In Sect. 80.2.1, we will describe the concept of Hamming code and show how to apply Hamming codes to data hiding. In Sect. 80.2.2, the basic theory and efficiency of "Hamming + 1" is presented.

80.2.1 Hamming Codes

Linear codes with length n and dimension k will be described as $[n, k]$ codes. Hamming codes are linear codes and will be 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 $c = \{xG|x \in F_q^k\}$. Matrix encoding conceals messages with the parity check matrix of linear codes. If $\boldsymbol{c}$ is an [n, k] linear code, the dual to it is an [n, $n - k$] linear code. If **H** is the checker matrix for $\boldsymbol{c}$, H is an $(n - k) \times k$ matrix the rows of which are orthogonal to $\boldsymbol{c}$ and $\{x \mid Hx^{\mathrm{T}} = 0\} = \boldsymbol{c}$.

$$(m_1, \ldots, m_k)^T = H \cdot (LSB(x_1), \ldots, LSB(x_n))^T \tag{80.1}$$

The Hamming codes function is to embed k bits $(m_1, \ldots, m_k) \in F_k^2$ in the LSBs of n pixel gray values $(x_1, \ldots, x_n)$ by at most R changes in the following manner. Note that the covering radius R is the largest number of possible changes and the purpose of Hamming codes is to minimize the average number of embedding changes R_a. In other words, the goal is to maximize the embedding efficiency k/R_a depending on the embedding rate k/n. We note that to correct one error, the position of the erroneous bit must be determined. For an n-bit code, $\log_2 n$ bits are therefore required. Equation (80.2) shows the parity check matrix for a (7, 4) Hamming code:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{80.2}$$

For c to be a codeword, it must be in the null space of this matrix, i.e., $Hc = 0$. Let us assume there is a sequence of bits that have an error in the first bit position, e.g., 1101010_b. We calculate the syndrome S with Eq. (80.1). c is a 7-bit binary number and T denote the transpose of a codeword c, that is, the syndrome is $([001])^{\mathrm{T}}$. A syndrome value that is not zero denotes the position of the erroneous bit. If one flips the bit at this position in the codeword, every bit of the codeword will be correct. Binary Hamming codes are [$2^r - 1$, $2^r - 1 - \mathrm{r}$] linear codes with a parity check matrix H of dimensions $r \times (2^r - 1)$ and whose columns are binary expansions of the numbers 1,…, $2^{\mathrm{r}} - 1$. For example, Eq. (80.3) shows the parity check matrix H for $r = 4$. Let us assume that the cover object is an image consisting of $P \times Q$ pixels.

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{80.3}$$

Example 1 We assume that the codeword $\boldsymbol{c}$ is [1101001]. It is easy to calculate the syndrome using Eq. (80.1) with the parity check matrix H and the codeword: $S = H \times (c)^T = ([000])^T$. If the computed syndrome vector S is 0, as in this case, there is no error in the codeword. Otherwise, there is an error in the bit at position S in c.

80.2.2 "Hamming + 1" Scheme

The parity check matrix of a Hamming code yields a covering function *COV* (1, $2^k - 1$, k), $k \geq 1$, i.e., embed k bits $(m_1, \ldots, m_k)$ into the LSBs of $2^k - 1$ pixel gray values $(x_1, \ldots, x_{2^k-1})$ using at most one change. This covering function is defined by $(m_1, \ldots, m_k)^T = H \cdot (LSB(x_1), \ldots, LSB(x_{2^k-1}))^T$, where H is the parity check matrix of [$2^k - 1$, $2^k - 1 - k$] Hamming code. Hamming covering function combines with one pixel to form "Hamming + 1" scheme, which embeds $k + 1$ bits into 2^k pixels gray values using at most one change:

$$(m_1, \ldots, m_k)^T = H \cdot (LSB(x_1), \ldots, LSB(x_{2^k-1}))^T \tag{80.4}$$

$$m_{k+1} = (\lfloor x_1/2 \rfloor + \cdots + \lfloor x_{2^k-1}/2 \rfloor + x_{2^k}) \bmod 2 \tag{80.5}$$

The first k bits are embedded into LSBs of the first $2^k - 1$ pixel values using the *COV* (1, $2^k - 1$, k) Hamming covering function, and the $(k + 1)$-*th* bit is a function of all 2^k pixels including the appended one. Note that by adding or subtracting one to/from a pixel value x, its $LSB(x)$ always becomes the same binary value $LSB(x) \oplus 1$, however, $\lfloor x/2 \rfloor \bmod 2$, which is the second least significant bit of x, can either be '0' or '1'. Therefore, when Eq. (80.4) does not hold, one pixel value, say, x_i, $1 \leq i \leq 2^k - 1$, has to be changed. By choosing $x_i + 1$ or $x_i - 1$, both Eqs. (80.4) and (80.5) can hold simultaneously without changing x_{2^k}.

On the other hand, when Eq. (80.4) holds but Eq. (80.5) does not, the first $2^k - 1$ pixels need not to be changed, and "Hamming + 1" scheme can modify x_{2^k-1} by randomly increasing or decreasing one to satisfy Eq. (80.5). This means that "Hamming + 1" scheme can embed $k + 1$ bits of message in 2^k pixels with at most one change. This method shows that the embedding efficiency is equal to $(k + 1)2^{k+1}/(2^{k+1} - 1)$, and the embedding rate is $(k + 1)/2^k$.

80.3 Proposed Method

This section proposes new data hiding method, which is called "Hamming + 3". Our proposed "Hamming + 3" improves the "Hamming + 1" scheme, which is a steganographic data hiding method, i.e., "Hamming + 1" scheme embeds $k + 1$ bits into 2^k pixels gray values using at most one change. Our proposed scheme improves the embedding rate compared to "Hamming + 1" scheme, i.e., "Hamming + 3" embeds $k + 3$ bits into $2^k - 1$ pixels gray values using at most 2 change.

80.3.1 “Hamming + 3” Scheme

We propose the following “Hamming + 3” scheme by appending three pixel after the block of Hamming covering function. It embeds $k + 3$ bits $(m_1, \ldots, m_k, m_{k+1}, m_{k+2}, m_{k+3})$ into $2^k - 1$ pixel gray values $(x_1, \ldots, x_{2^k})$ using at most two change:

$$(m_1, \ldots, m_k)^T = H \cdot (LSB(x_1), \ldots, LSB(x_{2^k}))^T \qquad (80.6)$$

$$(m_{k+1}, \ldots, m_{k+3})^T = H \cdot (\lfloor x_1/2 \rfloor \bmod 2, \ldots, + \cdots + \lfloor x_{2^k}/2 \rfloor \bmod 2) \qquad (80.7)$$

The first k bits are embedded into LSBs of the first $2^k - 1$ pixel values using the $COV\,(1, 2^k - 1, k)$ Hamming covering function [see Eq. (80.6)], and $k + 3$ bits are embedded into second least significant bits using the $COV\,(1, 2^k - 1, k)$ Hamming cover function [see Eq. (80.7)]. Therefore, when Eq. (80.6) does not hold, one pixel value, say, x_i, $1 \leq i \leq 2^k - 1$, has to be changed. By choosing $x_i + 1$ or $x_i - 1$, both Eqs. (80.6) and (80.7) can hold simultaneously without changing x_{2^k}. On the other hand, when Eq. (80.6) holds but Eq. (80.6) does not, the first $2^k - 1$ pixels need not to be changed, and “Hamming + 1” scheme can modify x_{2^k} by randomly increasing or decreasing one to satisfy Eq. (80.6). This means that “Hamming + 1” scheme can embed $k + 1$ bits of message in 2^k pixels with at most one change. This method shows that the embedding efficiency is equal to $(k + 1)2^{k+1}/(2^{k+1} - 1)$, and the embedding rate is $(k + 1)/2^k$. $[n, k]$ Hamming codes are now a linear space over a field of order q, prime. These q-ary codes are 1-error correcting, relying on the fact that each codeword is at a distance of at least 3 from any other codeword, which in turn relies on the construction of the matrix. Specifically, the fact that no two columns of the check matrix are linearly dependent means that the minimum distance between any two code-words is at least 3.

Proposition *Hamming Codes are 1-error correcting codes.*

Proof We need to check that

$$|C| \cdot \frac{1}{i = 0} \binom{x}{y} (q - 1)^i = |F_q|^n.$$

The right hand side of this is q^n, where $n = (q^r - 1)/(q - 1)$. The left hand side is

$$\begin{aligned} q^{n-r}(1 - n(q - 1)) &= q^{n-r}\left(1 + \frac{(q^r - 1)}{(q - 1)}(q - 1)\right) \\ &= q^{n-r}(1 + (q^r - 1)) \\ &= q^{n-r}(q^r) \\ &= q^n. \end{aligned}$$

80.3.2 Embedding 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 7-pixel blocks. We present the embedding procedure step by step:

Input: Cover image I sized $H \times W$, a binary secret message δ of maximum length $H \times W - 1$, and the parity check matrix H
Output: A stego image I' sized H × W

Step 1: Divide original images I into $1 \times 2^k - 1$ blocks, letting $c = (b(x_1), \ldots, b(x_{2^k-1}))$, where b(.) denote LSB of a pixel. Letting $c_2 = (\lfloor x_1/2 \rfloor \bmod 2, \ldots, \lfloor x_2^k/2 \rfloor \bmod 2)$. c and c_2 denote code-words and a set of LSB and second LSB respectively.

Step 2: Read all pixels and secret messages into array variable x and δ respectively. $CNT = \lfloor (H \times W)/7 \rfloor$.

Step 3: Calculate the syndrome S by applying Eq. (80.6) to the parity check matrix H and c, i.e., $S = H \cdot (b(x_i), \ldots, b(x_{2^k-1}))^T$, where $i = 1 \ldots n$. Compute $S_1 = S \oplus \delta_j^k$,, where $\oplus$ is XOR operation and $j = 1 \ldots n$, $j = j + k$. As S_1 is the position for 1-error correction, if S_1 is 0 then no flipping any pixel, else flipping a value of $b(x_{i+S_1})$.

Step 4: Calculate the syndrome S_2 by applying Eq. (80.7) to the parity check matrix H and c_2, i.e., $S_2 = H(\lfloor x_1/2 \rfloor \bmod 2, \ldots \lfloor x_{2^k}/2 \rfloor \bmod 2)^{\mathrm{T}}$, where $i = 1 \ldots n$, $i = i + (2^k - 1)$. Calculate the syndrome value for messages, $S_3 = S_2 \oplus \delta_j^k$, where $\oplus$ is XOR operation and $j = 1 \ldots n$, $j = j + k$. If S_3 is 0, then no flipping any pixel, else flipping a value of $(\lfloor x_{i+S_3}/2 \rfloor \bmod 2)$.

Step 5: Decrease CNT by $2^k - 1$. If CNT is greater than 0, return to step 3 to continue the process until there are no more pixels of I.

Example 2 A detailed explanation of the reasons is included in this example. A linear pixels $c = (67\ 79\ 83\ 88\ 91\ 93\ 95)$ is a $1 \times 2^k - 1$ block, reading from left to right and from top to bottom. The secret stream pixels are $\delta = (1\ 1\ 1\ 1\ 1\ 1)$, which is a $k + 3$ block. Calculate $S = (\mathrm{H} \cdot (\mathrm{b}(67)\ \mathrm{b}(79)\ \mathrm{b}(83)\ \mathrm{b}(88)\ \mathrm{b}(91)\ \mathrm{b}(93)\ \mathrm{b}(95))^T) \bmod 2 = (100)$. S is computation using LSB of a block of pixel. $S_2 = S \oplus \delta_j^k = (011)$. Compute $c_{D(S_2)} - 1 = 82$, where $D(.)$ is a function of binary-to-decimal conversion. Next, we show how to conceal secret bits k into second LSB layers. Calculate $S_3 = H \cdot ((\lfloor 67/2 \rfloor \lfloor 79/2 \rfloor \lfloor 83/2 \rfloor \lfloor 88/2 \rfloor \lfloor 91/2 \rfloor \lfloor 93/2 \rfloor \lfloor 95/2 \rfloor)^{\mathrm{T}} \bmod 2) = H \cdot (1\,1\,1\,0\,1\,0\,1) = (0\,1\,0)$. $\mathrm{S}_4 = \mathrm{S}_3 \oplus \delta_j^k = (1\,0\,1)$. Compute $c_{D(S_4)} - 2 = 89$.

80.3.3 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 7-pixel blocks. We present the extracting procedure step by step:

Input: Stego image I' sized $H \times W$ and the parity check matrix H
Output: A secret messages δ

Step 1: Divide stego images I' into $1 \times 2^k - 1$ blocks, letting $c = (\mathrm{b}(x_1), \ldots, \mathrm{b}(x_{2^k-1}))$, where $b(.)$ denote LSB of a pixel. Letting $c_2 = (\lfloor x_1/2 \rfloor \mathrm{mod}\ 2, \ldots, \lfloor x_{2^k}/2 \rfloor \ \mathrm{mod}\ 2)$. c and c_2 denote codewords and a set of LSB and second LSB, respectively.

Step 2: Read all pixels and secret messages into array variable x and δ respectively. $CNT = \lfloor (H \times W)/2 \rfloor$.

Step 3: Calculate the syndrome S by applying Eq. (80.6) to the parity check matrix H and c, i.e., $S = H \cdot (b(x_i), \ldots, b(x_{2^k-1}))^T$, where $i = 1 \ldots n$. Concatenate δ and S, i.e., $\delta = \delta || S$. A S denote extracted k bits.

Step 4: Calculate the syndrome S_1 by applying Eq. (80.7) to the parity check matrix H and c_2, i.e., $S_1 = \mathrm{H} \cdot (\lfloor x_1/2 \rfloor \mathrm{mod}\ 2, \ldots, \lfloor x_{2^k-1}/2 \rfloor \mathrm{mod}\ 2)^T$, where $i = 1 \ldots n$, $i = i + (2^k - 1)$. Concatenate δ and S_1, i.e., $\delta = \delta || S_1$. A S_1 denote extracted k bits. $j = 1 \ldots n, j = j + k$.

Step 5: Decrease CNT by $2^k - 1$. If CNT is greater than 0, return to Step 3 to continue the process until there are no more pixels of I.

80.4 Experimental Results

We proposed a "Hamming + 3" method for data hiding. To prove our proposed scheme is correct, we performed an experiment to verify that it ensures the hidden image can be restored. 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, 512 × 512 grayscale images were used as cover images. Figure 80.1 is a cover image for experiment to verify our proposed scheme. In our experiments, the qualities of the stego images are measured by the peak-signal-to-noise ratio (PSNR) [14].

The PSNR is the most popular criterion for measuring distortion between the original image and shadow images. It is defined as follows:

$$PSNR = 10 \times \log_{10}(255^2/MSE) \tag{80.8}$$

where MSE is the mean square error between the original grayscale image and the shadow image:

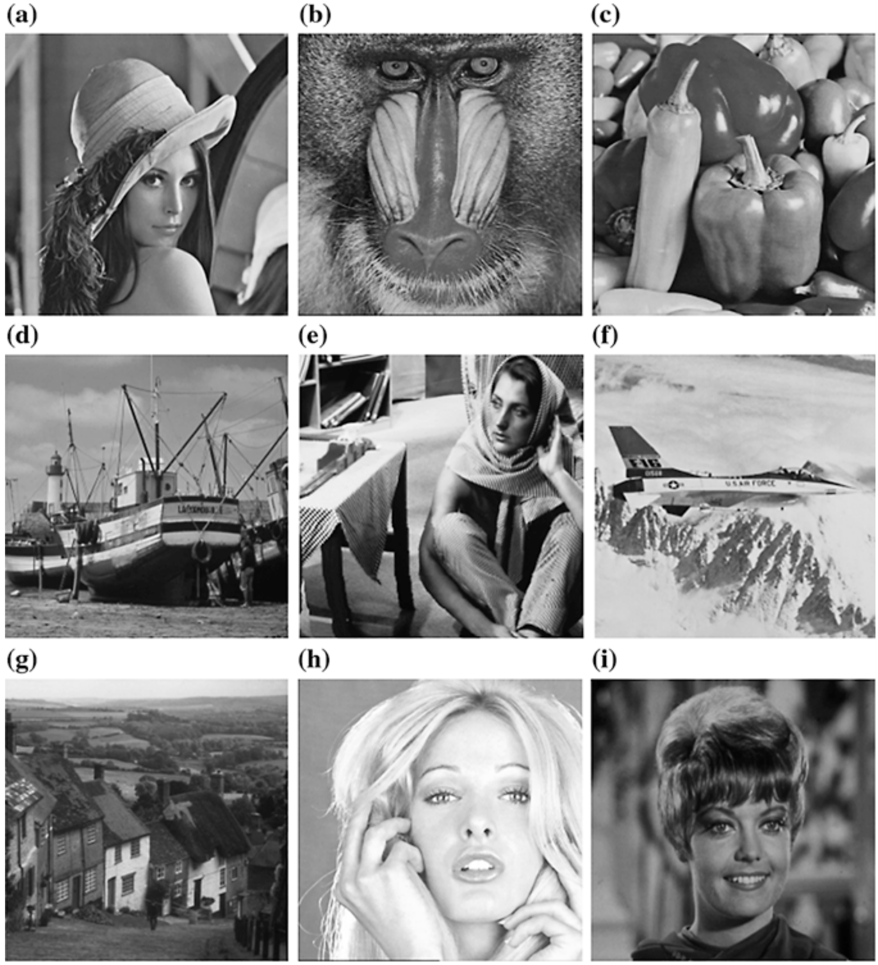

Fig. 80.1 512 × 512 grayscale cover images for data hiding experiment. **a** Lena. **b** Baboon. **c** Pepper. **d** Boat. **e** Barbara. **f** Airplane. **g** Goldhill. **h** Tiffany. **i** Zelda

$$MSE = \frac{1}{m \times n} \sum_{i}^{m} \sum_{j}^{n} \left[I(i,j) - I'(i,j) \right]^2 \tag{80.9}$$

The symbols $I(i, j)$ and $\mathrm{I}'(i, j)$ represent the pixel values of the original grayscale image and the stego image at position (i, j), respectively; m and n are the width and height of the original image, respectively.

$$p = \frac{|\delta|}{m \times n} (bpp) \tag{80.10}$$

Table 80.1 The comparison result of the matrix encoding, Hamming + 1 scheme and proposed scheme

Images	Method					
	Matrix coding		Hamming + 1		Hamming + 3	
	PSNR	Payload	PSNR	Payload	PSNR	Payload
Baboon	56.44	0.43	53.71	0.499	48.18	0.86
Barbara	54.65	0.43	48.60	0.499	48.22	0.86
Boats	54.75	0.43	49.37	0.499	48.20	0.86
Goldhill	57.02	0.43	53.73	0.499	48.21	0.86
Airplane	55.84	0.43	51.61	0.499	48.20	0.86
Lena	56.05	0.43	52.43	0.499	48.22	0.86
Pepper	54.01	0.43	47.26	0.499	48.22	0.86
Tiffany	53.40	0.43	47.46	0.499	48.20	0.86
Zelda	56.40	0.43	54.04	0.499	48.21	0.86
Average	56.44	0.43	50.91	0.499	48.20	0.86

In Eq. (80.10), p denotes bits-per-pixel (*bpp*), which is an embedding payload. Our experiment compares how many secret bits can be carried by a cover pixel. $|\delta|$ is the number of bits of a secret message δ. There is a tradeoff between a payload and quality of an image. To increase the embedding rate, it is too obvious to require a sacrifice of image quality.

However, if it is possible to keep the balance between payload and quality of an image, we then accomplish our purpose from an aspect of steganography. Table 80.1 shows the visual quality of the stego images created by the matrix encoding, "Hamming + 1", and "Hamming + 3". Our proposed "Hamming + 3" method shows 0.86 bpp with a good visual quality (i.e., the *PSNR* value is higher than 48 dB). From Table 80.1, for the visual quality factor, the matrix coding scheme shows a higher visual quality outcome.

For embedding payload comparison, the proposed "Hamming + 3" show a high embedding payload outcome. Although the visual quality of stego images generated by the "Hamming + 1" scheme is better than the proposed scheme, some images' quality were slightly lower than those of "Hamming + 3". In this experiment, we verified that "Hamming + 3" is worth the steganography method, because our scheme shows reasonable embedding rate and quality as a data hiding scheme. As the *PSNR* of our scheme is over 48 dB, it is not easily detectable by attackers. Therefore, our scheme is highly suitable for various fields of steganography.

80.5 Conclusion

In this paper, we proposed a "Hamming + 3" method that uses both layers, i.e., LSB and second LSB, using cover codes [n, k]. "Hamming + 1" can embed *COV* $(1, 2^k - 1, k)$ at the cost of $1/2^k + 1$ changes. The embedding efficiency is

$(k + 1)2^{k+1} - 1/(2^{k+1} - 1)$. Our proposed scheme shows 0.86 bpp, so “Hamming + 3” has better performance than “Hamming + 1”. Moreover, stego images of “Hamming + 3” are over 48 dB, and it denotes that our scheme is a reasonably acceptable steganography method. Thus, we can conclude that the “Hamming + 3” is suitable for steganographic applications.

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

References

1. Kim HJ, Kim C, Choi Y, Wang S, Zhang X (2010) Improved modification direction methods. Comput Math Appl 60(2):319–325
2. Yang CN, Ye G-C, Kim C (2011) Data hiding in halftone images by XOR block-wise operation with difference minimization. KSII Trans Internet Inf Syst 5(2):457–476
3. Crandall R (1998) Some notes on steganography. Posted on steganography mailing list, http://os.inf.tu-dresden.de/westfeld/crandall.pdf
4. Westfeld A (2001) F5: a steganographic algorithm. In: Proceedings of the 4th international workshop information hiding 2001. Lecture Notes in Computer Science, vol 2137, no 1, pp 289–302
5. Bierbrauer J (2005) Introduction to coding theory, Sect. 14.2. Chapman and Hall, CRC Press
6. Galand F, Kabatiansky G (2004) Information hiding by coverings. In: Proceedings of the IEEE information theory workshop 2004, pp 151–154
7. Bierbrauer J, Fridrich J (2006) Constructing good covering codes for applications in steganography. Available: http://www.math.mtu.edu/jbierbra/
8. Fridrich J, Soukal D (2006) Matrix embedding for large payloads. IEEE Trans Inf Secur Forensics 1(3):390–394
9. Schonfeld D, Winkler A (2006) Embedding with syndrome coding based on BCH codes. In: Proceedings of the 8th ACM workshop on multimedia and security, pp 214–223
10. Tseng Y-C, Chen Y–Y, Pan H-K (2002) A secure data hiding scheme for binary images. IEEE Trans Commun 50(8):1227–1231
11. Zhang X, Wang S (2006) Efficient steganographic embedding by exploiting modification direction. IEEE Commun Lett 10(11):781–783
12. Willems F, Dijk M (2005) Capacity and codes for embedding information in gray-scale signals. IEEE Trans Inf Theory 51(3):1209–1214
13. Mielikainen J (2006) LSB matching revisited. IEEE Signal Process Lett 13(5):285–287
14. Chang CC, Kieu TD, Chou YC (2008) A high payload steganographic scheme based on (7, 4) Hamming Code for digital images. In: International symposium on electronic commerce and security, pp 16–21