Sachnev and Kim *EURASIP Journal on Advances in Signal Processing* 2012, **2012**:89
http://asp.eurasipjournals.com/content/2012/1/89

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Modified BCH data hiding scheme for JPEG steganography

Vasily Sachnev[1] and Hyoung Joong Kim[2*]

Abstract

In this article, a new Bose-Chaudhuri-Hochquenghem (BCH)-based data hiding scheme for JPEG steganography is presented. Traditional data hiding approaches hide data into each block, where all the blocks are not overlapping each other. However, in the proposed method, two consecutive blocks can be overlapped to form a combined block which is larger than a single block, but smaller than two consecutive nonoverlapping blocks in size. In order to embed more amounts of data into the combined block than a single block, the BCH-based data hiding scheme has to be redesigned. In this article, we propose a way to get a joint solution for hiding data into two blocks with intersected coefficients such that any modification of the intersected area does not affect the data hiding process into both blocks. Due to hiding more amounts of data into the intersected area, embedding capacity is increased. On the other hand, the nonzero DCT coefficient stream is modified to achieve better steganalysis and to reduce the distortion impact after data hiding. This approach carefully inserts or removes 1 or -1 coefficients into or from the DCT coefficient stream according to the rule proposed in this article. Experimental results show that the proposed algorithms work well and their performance is significant.

Keywords: BCH, steganography, less detectable data hiding

1. Introduction

One of the first steganography methods for JPEG images embeds data by changing the least-significant bit values of the quantized discrete cosine transform (DCT) coefficients. However, this method can easily be detected by a statistical analysis. Thus, for a good while, evading the statistical analysis has been a major concern. Provos [1] divides the DCT coefficients into two disjoint subsets, hides data into the first subset, and compensates the distorted histogram by modifying the second subset. Other methods in [2,3] use a similar approach. On the other hand, Solanki et al. [4] utilize the robust watermarking scheme for steganography purposes. They embed data into image in the spatial domain by using a technique robust against JPEG compression. Their scheme provides less degradation onto the features of the DCT coefficients, and, as a result, its detectability was low against old version of the statistical steganalysis.

Another way to survive against steganalysis is reducing the number of modified coefficients. Traditionally, each nonzero DCT coefficient has been modified. As a result, embedding capacity is as much as the number of nonzero DCT coefficients. However, the maximum possible embedding capacity trades off the detectability. Westfeld [5] has used a matrix encoding (ME) technique to lower detectability by sacrificing the embedding capacity. The ME technique exploits the Hamming code which is designed for error correction. His scheme hides many bits by flipping at most one coefficient in each block. This approach was the first instance of using the error correcting code for data hiding.

Fridrich et al. [6-13] use the concept of the "minimal distortion" to enhance the security (i.e., by reducing distortion). The perturbed quantization steganography utilizes the wet paper coding.

Later, Kim et al. [14] have improved the performance of the ME by reducing the distortion impact. In fact, their modified matrix encoding (MME) method changes more number of coefficients compared to the ME. However, they show that the distortion impact after modifying one coefficient may be larger than that after modifying two coefficients. Thus, it is obvious that modifying one coefficient or two per block may have less distortion and

* Correspondence: khj-@korea.ac.kr
[2]CIST, Korea University, Seoul 136-701, Republic of Korea
Full list of author information is available at the end of the article

lower detectability against the steganalysis. Note that MME requires the original uncompressed image for data hiding, but not for decoding.

Schönfeld and Winkler [15] have proposed a new way to hide data using more powerful error correction code. They use a structured Bose-Chaudhuri-Hochquenghem (BCH) code [2]. Zhang et al. [16] have significantly improved the original BCH-based data hiding scheme. Their improved method can easily find the flip positions and defeat the steganalysis well compared to the existing methods. Later, Sachnev et al. [17] apply a heuristic optimization technique for the data hiding scheme over the BCH coding and modify the stream of the input DCT coefficients to reduce the distortion. Their method considerably outperforms the steganography method proposed by Zhang et al. [16].

Recently, Filler and Fridrich [18] have proposed a remarkable framework which minimizes a distortion measure as a weighted norm of the difference between cover and stego feature vectors. In their approach, the distortion is not necessarily an additive function over the pixels because the features may contain higher-order statistics such as sample transition probability matrices of pixels or DCT coefficients modeled as Markov chains [19-21]. When the distortion measure is defined as a sum of local potentials, practical near-optimal embedding methods can be implemented with syndrome-trellis codes [22].

Most of the above-mentioned steganographic methods use the nonoverlapping blocks of the DCT coefficients for hiding secret message. Such a blockwise embedding scheme divides both the stream of the DCT coefficients and hidden message into the separate blocks and solves the equations for hiding data for each block individually. Recent methods like MME [14], BCH-based steganography methods [15-17] may produce several alternative solutions. Thus, such a data hiding method can choose a solution with the lowest distortion impact. Past investigation over the BCH data hiding scheme finds that BCH usually allows redundant number of possible solutions. It means that a solution with acceptable distortion impact can be achieved from the reduced set of possible solutions. Hence, the embedding efficiency of the BCH steganographic methods can be increased by reducing the number of possible solutions and keeping similar distortion impact compared to the original approach.

In the proposed method, two blocks of the DCT coefficients form a combined block sharing common coefficients in the intersected part between two consecutive blocks. Such a design achieves high embedding efficiency by hiding data twice into the intersected area. The number of possible joint solutions for both blocks (i.e., solutions which valid for both blocks) is always smaller than the number of all possible solutions for two independent blocks. The reduced number of possible solutions can increase distortion, but not significantly. Besides, the number of possible solutions can easily be controlled by changing size of the intersected area. The smaller size of the intersected area, the larger number of possible joint solutions. Similar approach has been tested for Hamming code in [23].

However, the higher size of the intersected area, the higher embedding efficiency of the proposed method. In the proposed method, the block of the DCT coefficients can be modified by inserting new nonzero coefficients 1 or -1, or removing coefficients 1 or -1. Such modification is carried out carefully and sophisticatedly in order to reduce distortion caused by excessive hiding.

The rest of the article is organized as follows. Section 2 explains the details of the BCH coding. Section 3 presents the BCH-based modified data hiding scheme. In Section 4, we propose the inserting-removing strategy. The encoder and decoder are presented in Section 5. Section 6 provides the experimental results. Finally, Section 7 concludes the article.

2. BCH syndrome coding

The BCH codes are the well known and widely used family of the error correction codes. BCH code (*n, k, t*) can correct *t* bits by inserting *n* - *k* additional bits to the original message *k* such that syndrome of resulted *n* bits is equal to 0. In general, BCH codes were invented for error correction and cannot directly be used for data hiding. An efficient method of using powerful BCH codes for data hiding has been presented in [15-17].

2.1. BCH syndrome coding

The generalized parity-check matrix *H* for BCH coding is presented as follows:

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \left(\alpha^3\right) & \left(\alpha^3\right)^2 & \cdots & \left(\alpha^3\right)^{n-1} \\ \vdots & & & & \vdots \\ 1 & \left(\alpha^{2t-1}\right) & \left(\alpha^{2t-1}\right)^2 & \cdots & \left(\alpha^{2t-1}\right)^{n-1} \end{bmatrix} \quad (1)$$

Let *t* be 2. Then, the parity-check matrix is expressed as follows:

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \left(\alpha^3\right) & \left(\alpha^3\right)^2 & \cdots & \left(\alpha^3\right)^{n-1} \end{bmatrix} \quad (2)$$

Assume that the original stream of binary data is $\mathbf{V} = \{v_0, v_1, v_2, \ldots, v_{n-1}\}$, and the modified stream of binary data after data hiding is $\mathbf{R} = \{r_0, r_1, r_2, \ldots, r_{n-1}\}$. The streams **V** and **R** over $GF(2^m)$ can be represented as $\mathbf{V}(x) = v_0 + v_1 \cdot x + v_2 \cdot x^2 + v_3 \cdot x^3 + \ldots + v_{n-1} \cdot x^{n-1}$, and $\mathbf{R}(x) = r_0 + r_1 \cdot x + r_2 \cdot x^2 + r_3 \cdot x^3 + \ldots + r_{n-1} \cdot x^{n-1}$, respectively.

The embedded message $\boldsymbol{m}$ can be computed as follows:

$$\boldsymbol{m} = \mathbf{R} \cdot \mathbf{H}^T \quad (3)$$

Thus, the hiding message $\mathbf{m}$ to $\mathbf{V}$ requires to find $\mathbf{R}$ such that

$$\mathbf{R} \cdot \mathbf{H}^T = \boldsymbol{m} \quad (4)$$

The difference between $\mathbf{V}$ and $\mathbf{R}$ shows the number and location of the elements in $\mathbf{V}$ to be flipped.

$$\mathbf{R} = \mathbf{V} + \mathbf{E} \quad (5)$$

or

$$\mathbf{E} = x^{u_1} + x^{u_2} + x^{u_3} + \cdots + x^{u_l},$$

where $u = \{u_0, u_1, u_2, ..., u_l\}$ are the positions of the elements in $\mathbf{V}$ to be flipped in order to get $\mathbf{R}$.

Using Equations (3) and (4), the syndrome $\mathbf{S}$ can be computed as follows:

$$\boldsymbol{S} = \boldsymbol{m} - \mathbf{V} \cdot \mathbf{H}^T = \mathbf{E} \cdot \mathbf{H}^T. \quad (6)$$

If t is 2, then

$$\boldsymbol{S} = \left[S_1 \ S_2\right]^{\mathrm{T}} = \mathbf{E} \cdot \mathbf{H}^T. \quad (7)$$

2.2. Lookup tables

In this article, we utilized the method of Zhao et al. [24] based on the fast lookup tables for finding roots of quadratic and cubic polynomial of $\sigma(x)$. Similar approach has been used in [16,17].

2.3. Solutions

Hiding message $\boldsymbol{m}$ to the binary stream V requires to find the positions of the coefficients to be flipped. In this article, we used a method presented in [16,17] to get one, two, three, or four flips solutions. The set of all possible solutions for one, two, three, or four flips has to be stored in the look up tables J_1, J_2, J_3, and J_4, respectively. The notation $J_3(S)$ returns all three flips solutions for syndrome $S = \{S_1 \ S_2\}$. Similarly, we can get all possible solutions for block n_1 with syndrome S^{I}, for block n_2 with syndrome S^{II}, as $J^{\mathrm{I}} = \{J_1(S^{\mathrm{I}}) \ J_2(S^{\mathrm{I}}) \ J_3(S^{\mathrm{I}}) \ J_4(S^{\mathrm{I}})\}$ and for block n_2 with syndrome S^{II} as $J^{\mathrm{II}} = \{J_1(S^{\mathrm{II}}) \ J_2(S^{\mathrm{II}}) \ J_3(S^{\mathrm{II}}) \ J_4(S^{\mathrm{II}})\}$, respectively. The look up tables' size is $(2^{2 \cdot m} - 1) \times$ nS where nS is a number of stored solutions.

3. Proposed data hiding scheme

In the proposed BCH data hiding scheme, we combine two BCH blocks of $2^{\mathrm{m}} - 1$ DCT coefficients into one, such that BCH blocks intersect each other. Figure 1 shows the block diagram of coefficients for the proposed scheme. In the presented example, $(a_1, a_2, a_3, ..., a_{25})$ is the combined block of the DCT coefficients; $(v'_1, v'_2, v'_3, \ldots, v'_{15})$ and $(v''_1, v''_2, v''_3, \ldots, v''_{15})$ are the corresponding binary coefficients for the BCH blocks n_1 and n_2, respectively. Intersected area I covers five coefficients a_{11}, a_{12}, a_{13}, a_{14}, and a_{15} in this example. Such a scheme can hide more amounts of data by exploiting the intersected area using any kind of coding schemes.

One of the two main contributions of this article is to present a systematic algorithm for the joint solutions. The proposed BCH-based data hiding scheme requires to find a joint solution for both blocks n_1 and n_2 using the guidelines from Section 2.1 such that the intersected area does not affect the result. For example, let 8 bits be hidden into 15 coefficients from a_1 to a_{15} using the BCH-based steganography. Then, another 8 bits can be hidden into the next block having another 15 coefficients from a_{11} to a_{25}. This is the traditional approach. As a result, 16 bits can be hidden into 30 coefficients. However, our new approach hides the same amount of data into 25 coefficients a_1 to a_{25}. Eight bits are hidden into the coefficients from a_1 to a_{15}, and another eight bits into the coefficients from a_{11} to a_{25}. Data hiding algorithm requires to find syndromes S^{I} and S^{II} (Equation 6) for each block n_1 and n_2, respectively.

There are two possible ways for hiding data into the combined blocks. Either hiding data into the block n_1 first, or into the block n_2 first. The proposed algorithm for getting a joint solution is designed as follows:

1. Hiding data into the block n_2 first.
   (a) Some solutions for hiding data into the block n_1 do not modify the coefficients in the

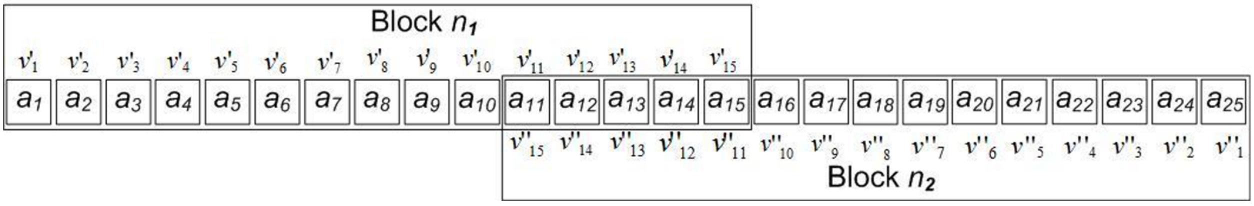
Block n_1
v'_1 v'_2 v'_3 v'_4 v'_5 v'_6 v'_7 v'_8 v'_9 v'_{10} v'_{11} v'_{12} v'_{13} v'_{14} v'_{15}
a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} a_{13} a_{14} a_{15} a_{16} a_{17} a_{18} a_{19} a_{20} a_{21} a_{22} a_{23} a_{24} a_{25}
v''_{15} v''_{14} v''_{13} v''_{12} v''_{11} v''_{10} v''_9 v''_8 v''_7 v''_6 v''_5 v''_4 v''_3 v''_2 v''_1
Block n_2

Figure 1 Two intersected blocks of the modified BCH data hiding scheme.

intersected area. Thus, solutions for the block n_2 have to be obtained using the original syndrome S^{II}. Some solutions are valid since they do not modify the coefficients in the intersected area. These solutions are called specified solutions.

(b) Some solutions for the block n_1 modify the coefficients in the intersected area. These modifications in the intersected area affect the syndrome for the block n_2. Thus, the new syndrome for the block n_2 is obtained as S^{II} new. Some new solutions are valid since they do not modify the coefficients already modified by the n_1 in the intersected area.

Among all possible solutions for the block n_2 and new syndrome S^{II}_{new} (in case of 1(a), $S^{\mathrm{II}}_{new} = S^{\mathrm{II}}$), choose the solutions which do not have flipping positions in the intersected area (i.e., valid or specified solutions). Thus, the joint solutions for a combined block unify the solutions for the block n_1 and its syndrome S^{I} and the specified solutions for the block n_2 and its syndrome S^{II} new.

2. Hiding data into the block n_2 first.

   (a) Some solutions for hiding data into the block n_2 do not modify the coefficients in the intersected area. Thus, solutions for the block n_1 have to be obtained using the original syndrome S^{I} . Some solutions are valid since they do not modify the coefficients in the intersected area.

   (b) Some solutions for the block n_2 modify the coefficients in the intersected area. These modifications in the intersected area affect the syndromes for the block n_1. Thus, the new syndrome for the block n_1 is obtained as S^{I}_{new}. Some new solutions are valid since they do not modify the coefficients already modified by the n_2 in the intersected area.

The joint solutions for a combined block unify the solutions for the block n_2 and its syndrome S^{II} and the specified solutions for the block n_1 and its syndrome S^{I} new (in case of 2(a), $S^{\mathrm{I}}_{new} = S^{\mathrm{I}}$).

In general, the proposed modified BCH data hiding schemes hides 4·m bits of data to the block of 2·(2^m-1)-|I| by using the BCH scheme (2^m-1, k, 2) for blocks n_1 and n_2.

The proper BCH-based data hiding scheme needs a suitable parameter m for hiding message M into the stream of N nonzero DCT coefficients. The parameter m can be obtained as follows:

$$\frac{4 \cdot m \cdot N}{2 \cdot (2^m - 1) - |I|} \geq |M|, \tag{8}$$

where m defines the proper BCH-based scheme for the proposed method, N is the number of nonzero DCT coefficients, M is the hidden message, n^p = 2·(2^m-1)-|I| is the size of the combined block, 4·m is the capacity of the combined block.

3.1. Data hiding algorithm

The proposed method requires to find the solution for two blocks n_1 and n_2 for hiding two messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ together such that

$$\begin{cases} \mathbf{m}_1 = \mathbf{H} \cdot \mathbf{R}_1 \\ \mathbf{m}_2 = \mathbf{H} \cdot \mathbf{R}_2 \end{cases} \tag{9}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are the modified streams of the binary coefficients obtained from n_1 and n_2 (see Figure 1); $\mathbf{H}$ is a parity-check matrix from Equation (1).

Note that, hiding message $\boldsymbol{m}_1$ to block n_1 modifies the block n_2 and vice versa, due to the intersected part. Hence, we need proper positions to flip by solving Equation (9) for correct decoding.

Among all possible solutions, the proposed method unifies the solutions for blocks n_1 and n_2, such that the flip positions cover only nonintersected area for both blocks (i.e., $J^{\mathrm{I}}_s = J^{\mathrm{I}} \notin I$ and $J^{\mathrm{II}}_s = J^{\mathrm{II}} \notin I$, for blocks n_1 and n_2). In other words, it is desirable to hide data into the block n_1 using the solutions from J^{I}_s that do not affect the block n_2, and vice versa. According to the above explanation, J^{I}_s and J^{II}_s unify the specified solutions for the blocks n_1 and n_2, respectively. Here, note that superscript indexes X^{I} and X^{II} present different items for blocks n_1 and n_2, respectively.

However, even if some flip positions j from the block n_1 belong to the intersected area I. Thus, we can consider the effect of those j to get a new solutions for the block n_2 and vice versa.

For this purpose, Equation (9) can be rewritten as follows:

$$\begin{array}{l} P^{\mathrm{II}}_1 = S^{\mathrm{II}}_1 + \beta_1 + \ldots + \beta_l \\ P^{\mathrm{II}}_2 = S^{\mathrm{II}}_2 + \beta^3_1 + \ldots + \beta^3_l \end{array} \tag{10}$$

where $S^{\mathrm{II}} = \left\{ S^{\mathrm{II}}_1 \; S^{\mathrm{II}}_2 \right\}$ is the syndrome for blocks n_2; $S^{\mathrm{II}}_{new} = \left\{ P^{\mathrm{II}}_1 \; P^{\mathrm{II}}_2 \right\}$ is a new syndrome for blocks n_2 after hiding data to block n_1; l is the number of the flip positions (j_1, ..., j_l) from the block n_1 belonged to the intersected area I (i.e., j_1, ..., j_l = $J^I(S^I) \in I$); and the values β_1, ..., β_l are computed using Equation (13) for the flipping positions $(j'_1, \ldots, j'_l) = F(j_1, \ldots, j_l)$ from the intersected area I for the block n_2. Function F converts indexes (j_1, ..., j_l) of the intersected area from the block n_1 to the corresponding indexes $(j'_1, \ldots, j'_l)$ from the block n_2. For example, solution for the block n_1 illustrated in Figure 1 is $J^{\mathrm{I}}\left(S^{\mathrm{I}}\right)=\left[3\ 11\right]$. $j_1 = 11 \in I$, where index 1 means the first coefficient form the intersected area I. Coefficient $j_1 = 11$ is located in the 11th position of the combined block. However, 11th coefficient in the combined block is the 15th coefficient in the block n_2 (i.e., $F\left(j_1\right) = F(11) = j'_1 = 15$ see Figure 1). Thus, even if the flip positions for blocks n_1 and n_2 are different (i.e., $j_1 = 11$ and $j'_1 = 15$), those coefficients have the same location in the combined block.

Finally, the solution for the block n_2 can be obtained as $\{j'_1, \ldots, j'_l, J_s^{\mathrm{II}}(S_{new}^{\mathrm{II}})\}$. Presented solution sufficiently hides message $\boldsymbol{m}_2$ into the block n_2.

The joint solution hides both messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ into the combined blocks. The joint solution $\left\{J^{\mathrm{I}}\left(S^{\mathrm{I}}\right),\ J_s^{\mathrm{II}}\left(S_{new}^{\mathrm{II}}\right)\right\}$ unifies the solutions for the blocks n_1 and n_2. In this example, the flipping positions from the intersected area are the part of $J^{\mathrm{I}}(S^{\mathrm{I}})$.

Similarly, we can get a joint solution by using the current solution for block n_2 (i.e., $J^{\mathrm{II}}(S^{\mathrm{II}})$). For this purpose, Equation (9) can be rewritten again as follows:

$$\begin{array}{l} P_1^{\mathrm{I}} = S_1^{\mathrm{I}} + \beta_1 + \ldots + \beta_l \\ P_2^{\mathrm{I}} = S_2^{\mathrm{I}} + \beta_1^3 + \ldots + \beta_l^3 \end{array} \tag{11}$$

where $S^{\mathrm{I}} = \left\{S_1^{\mathrm{I}}\ S_2^{\mathrm{I}}\right\}$ is the syndrome for blocks n_1; $S_{new}^{\mathrm{II}} = \left\{P_1^{\mathrm{II}}\ P_2^{\mathrm{II}}\right\}$ is the new syndrome of the block n_1 after hiding data to block n_2; l is the number of flip positions $(j'_1, \ldots, j'_l)$ for the block n_2 belonged to the intersected area I (i.e., $(j'_1, \ldots, j'_l) = J^{\mathrm{II}}(S^{\mathrm{II}}) \in I$); $\beta_1, \ldots, \beta_l$ are computed using Equation (15) for the flipping positions $(j_1, \ldots, j_l) = F^{-1}(j'_1, \ldots, j'_l)$ from the intersected area I for the block n_1; function F^{-1} (i.e., the inverse function of F) converts the indexes of the coefficients of intersected area $(j'_1, \ldots, j'_l)$ from the block n_2 to the corresponding indexes $(j_1, \ldots, j_l)$ from the block n_1. For example, if $J^{\mathrm{II}}\left(S^{\mathrm{II}}\right) = \left[1\ 15\right]$, then $j'_1 = 15 \in I$, then $j_1 = F^{-1}\left(j'_1\right) = F^{-1}(15) = 11$ (see Figure 1).

The solution for the block n_1 can be obtained as $\left\{(j_1, \ldots, j_l)\, J_s^{\mathrm{I}}(S_{new}^{\mathrm{I}})\right\}$. Presented solution sufficiently hides message $\boldsymbol{m}_1$ into the block n_1.

Joint solution for hiding both messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ is $\left\{J_s^{\mathrm{I}}(S_{new}^{\mathrm{I}})\, J^{\mathrm{II}}(S^{\mathrm{II}})\right\}$. Here, the flipping positions from the intersected area $(j'_1, \ldots, j'_l)$ are the part of $J^{\mathrm{II}}(S^{\mathrm{II}})$. Corresponding flipping positions $(j_1, \ldots, j_l) = F^{-1}(j'_1, \ldots, j'_l)$ are the part of the solution for the block n_1.

Note that there are several solutions in J^{I} and J^{II} for syndromes S^{I} and S^{II}, respectively. Presented method may generate one joint solution for each solution from $J^{\mathrm{I}}(S^{\mathrm{I}})$ and $J^{\mathrm{II}}(S^{\mathrm{II}})$.

The proposed method requires to find values β from the flip positions $(j_1, \ldots, j_l)$ or $(j'_1, \ldots, j'_l)$. The relationship between β and flip position j is presented as follows:

$$j = \log(\beta) \tag{12}$$

or

$$\beta = \log^{-1}(j) \tag{13}$$

The complete procedure for getting all possible joint solutions for any syndromes is presented as follows:

For a given combined block of binary coefficients a and two messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ process follows:

(a) Define two blocks of the DCT coefficients n_1 and n_2 (see Figure 1). Compute syndromes S^{I} and S^{II} using corresponding binary streams *v'* and *v“*.
(b) Find all possible solutions $J^{\mathrm{I}} = J^{\mathrm{I}}(S^{\mathrm{I}})$ and $J^{\mathrm{II}} = J^{\mathrm{II}}$ (S^{II}) for blocks n_1 and n_2 by using the syndromes S^{I} and S^{II}.
(c) For each solution $J^{\mathrm{I}}(p)$ (p = 1, 2, 3,..,k, where k is the number of solutions) process follows:
i. Define flip positions $j_1, \ldots, j_l$ from the intersected area I.
ii. Convert $j_1, \ldots, j_l$ to $j'_1, \ldots, j'_l$ (corresponding flip positions from the block n_2). Compute corresponding β using Equation 13. Compute new syndrome S_{new}^{II} using Equation 10.
iii. Using a new syndrome S_{new}^{II} get new flips solutions as $J_{new}^{\mathrm{II}} = J_s^{\mathrm{II}}\left(S_{new}^{\mathrm{II}}\right)$.
iv. For each solution $J_{new}^{\mathrm{II}}(q)$ (q = 1, 2, 3,...,z, where z is the number of solutions) store the joint solution: $\left\{J^{\mathrm{I}}\left(p\right), J_{new}^{\mathrm{II}}(q)\right\}$.
(d) For each solution J^{II} (p) (p = 1, 2, 3,...,k) process follows:
i. Define flip positions $j'_1, \ldots, j'_l$ from the intersected area I for block n_2.
ii. Convert $j'_1, \ldots, j'_l$ to $j_1, \ldots, j_l$ Compute corresponding β. Compute new syndrome S_{new}^{I} using Equation 11.
iii. Using a new syndrome S_{new}^{I} get new flips solutions as $J_{new}^{\mathrm{I}} = J_s^{\mathrm{I}}\left(S_{new}^{\mathrm{I}}\right)$.
iv. For each solution $J_{new}^{\mathrm{I}}(q)$ (q = 1, 2, 3,...,z, where z is the number of solutions) store the joint solution: $\left\{J_{new}^{\mathrm{I}}\left(q\right), J^{\mathrm{II}}(p)\right\}$.

The stored joint solutions are used further to hide data with better performance. Note that the proposed method needs to search the best solution among $k{\cdot}q$

possible candidates for each block (see steps c and d). Thus, computational complexity of the proposed search algorithm is $O(n^2)$.

3.2. Two-stage embedding technique

In order to enhance the performance of the blockwise methods (i.e., ME, MME, BCH-based data hiding, etc.), we utilize almost all the DCT coefficients for data hiding. The proposed method uses two different embedding schemes together. Two schemes use the different block sizes n_1^p and n_2^p, and have different payloads m_1^p and m_2^p.

This method divides the stream of the DCT coefficients (c_1, c_2, ..., c_N) and the message M into two parts and hides data into each part separately. The optimal number of the blocks (k_1 and k_2) for both schemes can be computed as follows:

The relation between the numbers of blocks for the schemes 1 and 2 is presented as follows:

$$\begin{cases} n_1^p \cdot k'_1 + n_2^p \cdot k'_2 = N \\ m_1^p \cdot k'_1 + m_2^p \cdot k'_2 = |M| \end{cases} \tag{14}$$

where N is the number of DCT coefficients.

The computed k'_1 and k'_2 are noninteger numbers. Thus, we have to choose the nearest integers $k_1 = \lceil k'_1 \rceil \pm 1$ and $k_2 = \lceil k'_2 \rceil \pm 1$ such that:

$$\begin{cases} n_1^p \cdot k_1 + n_2^p \cdot k_2 \leq N \\ m_1^p \cdot k_1 + m_2^p \cdot k_2 \geq |M| \end{cases} \tag{15}$$

The presented two-scheme embedding method improves the performance of data hiding by using the proper distribution of the available DCT coefficients among two different modified BCH schemes. First scheme uses $m_1^p = 4 \cdot m$ obtained from inequality (8), the second scheme uses $m_p^2 = 4 \cdot (m + 1)$. Note that the second scheme has higher embedding efficiency. The efficiency of the two schemes embedding refers to the ratio between number of blocks k_1 and k_2 for the schemes 1 and 2, respectively. The larger the value k_1 (smaller ratio k_1/k_2), the higher efficiency of the proposed two schemes embedding for the same m.

The two-scheme embedding method enables to use different sizes of the intersected area for both schemes I_{sh1} and I_{sh2}, respectively (see Tables 1 and 2 We test several sizes of the intersected areas and several payloads. In the experiments, we try to hide data into a set of 4,000 natural images and compute performance against the steganalysis [20,25] for different sizes of the intersected areas and payloads. Results are presented in Tables 1, 2, and 3.

Table 1 Accuracy of the steganalysis [20] for different sizes of the intersected areas and payloads

Payload bpc	Accuracy of the stege analysis					
	Proposed method					BCH[16] \improvement
0.05			I_{sh2}			50.12\0.09
			10%	30%	50%	
	I_{sh1}	30%	50.11	50.08	50.04	
		50%	50.05	50.06	**50.03**	
0.1			I_{sh2}			51.54\1.51
			10%	30%	50%	
	I_{sh1}	30%	50.11	50.06	50.04	
		40%	50.05	50.07	**50.03**	
0.15			I_{sh2}			57.13\6.58
			30%	35%	40%	
	I_{sh1}	10%	**50.25**	50.31	50.48	
		15%	50.29	50.28	50.55	
0.17			I_{sh2}			60.03\7.22
			25%	35%	45%	
	I_{sh1}	5%	53.87	54.01	53.96	
		15%	53.10	**52.28**	52.81	
		30%	53.91	54.12	53.89	
0.2			I_{sh2}			65.54\6.62
			10%	30%	50%	
	I_{sh1}	5%	59.81	59.51	60.02	
		15%	59.26	**58.19**	59.01	
		30%	59.98	60.11	59.45	
0.22			I_{sh2}			73.06\10.6
			25%	35%	50%	
	I_{sh1}	30%	62.18	**62.01**	62.24	
		50%	65.21	65.53	65.31	
0.25			I_{sh2}			80.45\11.33
			30%	40%	50%	
	I_{sh1}	50%	69.38	69.25	**69.13**	

where I_{sh1} and I_{sh2} are the intersected area size in terms of the percent point from the block size 2^m - 1 for schemes m_p^1 and m_p^2

Boldface numbers in Tables 1 and 2 link to the lowest accuracy and show the most appropriate intersected area size for each tested payload. Data hiding by using the most appropriate intersected area always shows better results. Tables 1 and 2 also indicate a difference between the proposed method and the original BCH-based steganography method [16] in terms of performance of the steganalysis [20,25]. The most appropriate intersected area size presented in Table 3 was used later for other experiments.

4. Inserting-removing strategy

The performance of the proposed method can significantly be increased by using inserting-removing strategy.

Table 2 Accuracy of the steganalysis [25] for different sizes of the intersected areas and payloads

Payload bpc	Accuracy of the stege analysis					
	Proposed method					BCH[16] \improvement
0.05			l_{sh2}			50.12\0.1
			10%	30%	50%	
	l_{sh1}	30%	50.13	50.08	50.10	
		50%	50.06	50.05	**50.02**	
0.1			l_{sh2}			51.54\1.48
			10%	30%	50%	
	l_{sh1}	30%	50.10	50.07	50.08	
		50%	50.09	50.11	**50.06**	
0.15			l_{sh2}			57.03\3.92
			30%	35%	40%	
	l_{sh1}	10%	**53.11**	53.01	52.89	
		15%	52.71	52.88	52.78	
0.17			l_{sh2}			60.34\3.29
			25%	35%	45%	
	l_{sh1}	5%	57.90	57.28	57.61	
		15%	57.46	**57.05**	57.82	
		30%	57.95	58.10	58.14	
0.2			l_{sh2}			66.43\3.38
			10%	30%	50%	
	l_{sh1}	5%	63.88	63.57	64.21	
		15%	63.51	**63.05**	64.18	
		30%	64.22	64.30	64.12	
0.22			l_{sh2}			75.15\7.65
			25%	35%	50%	
	l_{sh1}	30%	67.94	**67.50**	67.82	
		50%	68.33	68.52	68.12	
0.25			l_{sh2}			82.79\8.39
			30%	40%	50%	
	l_{sh1}	50%	74.28	74.38	**74.40**	

The proposed strategy is based on fact that the block of the 2^m -1 DCT coefficients can be modified before data hiding by inserting or removing coefficients 1 and -1. Data hiding to modified stream of DCT coefficients may result lower distortion and, as a result, lower detectability of the steganalysis. Such a modification has to be carried out carefully and sophisticatedly in order to reduce distortion.

The proposed inserting-removing strategy uses the stream of nonrounded quantized DCT coefficients a_q computed as follows:

$$a' = DCT\,(B)\,, \quad a_q = \frac{a'}{Q}, \quad a_r = round(a_q) \tag{16}$$

where B is the 8 × 8 block of the image pixels; a' is the block of original DCT coefficients; a_q is the block of DCT coefficients divided by corresponding coefficients from quantization matrix Q; a_r is the block of quantized DCT coefficients; Q_f is a quality factor.

Each nonzero integer DCT coefficient has a corresponding informative bit computed as follows:

$$b = \begin{cases} a_r mod2 & if\, a_r > 0, \\ a_r - 1\, mod2 & if\, a_r < 0 \end{cases} \tag{17}$$

According to the proposed inserting-removing strategy, the stream a of nonrounded DCT coefficients obtained from the blocks a_q is divided into three sets: modifiable $c_m = a \in (-\infty; -1.5) \cup (1.5;\infty)$, removable $c_R = a \in [-1.5; -0.5) \cup (0.5;1.5]$, and insertable $c_{Ins} = a \in [-0.5; -0.25) \cup (0.25;0.5]$. Set c unifies modifiable, insertable, and removable sets (i.e., $c = c_m \cup c_R \cup c_{Ins}$). The set $C = c_m \cup c_R$ contains all nonzero rounded DCT coefficients. According to Equation (17), only the nonzero DCT coefficients (i.e., set C) have the corresponding informative coefficients and can be used for hiding data.

The proposed steganographic method uses the stream of n_p nonzero DCT coefficients from the set C for data hiding. In general, set C is the subset of the unified set c. Thus, each block unifies the n_p coefficients form set C and some insertable coefficients from the set c (i.e., $c_b = c'_m \cup c'_R \cup c'_{Ins}$, where $C' = c'_m \cup c'_R$ is the block of n_p nonzero DCT coefficients from the set C). Inserting or removing of any coefficients from c'_{Ins} and c'_R produces a new block C' with new solution for data hiding. As a result, inserting-removing strategy significantly increases the number of possible solutions and helps to find the most appropriate solution with the lowest distortion.

In the proposed improved matrix encoding, we use the same measure for computing distortion similar to MME [14]. The distortion for each DCT coefficient is computed as follows:

$$D = E^2 \cdot Q^2 \tag{18}$$

$$E = \begin{cases} 0.5 - |C - \lfloor C \rceil|\,, & \text{if}\, C \in c_m \\ 1.5 - |C|\,, & \text{if}\, C \in c_R \end{cases}$$

Table 3 The most appropriate intersected area size versus payload

	Payload size (bit per nonzero coefficient)						
	0.05 (%)	**0.1 (%)**	**0.15 (%)**	**0.17 (%)**	**0.2 (%)**	**0.22 (%)**	**0.25 (%)**
Scheme m_p^1	50	50	10	15	15	30	50
Scheme m_p^2	50	50	30	35	30	35	50

The distortion due to inserting or removing D_{IR} is computed as follows:

$$D_{IR} = |0.5 - |C||^2 \cdot Q^2, \quad if\ C \in c_R \cup c_{Ins} \tag{19}$$

where Q is the corresponding quantization coefficient of the quantization table.

The resulted distortion for the combined block of DCT coefficients is computed as follows:

$$D_b = \sum_{i=1}^{l} D_i + D_{IR} \tag{20}$$

where l is the number of flipped coefficients.

Flipped coefficients are computed as follows:

$$A_r = \begin{cases} 2, & if\ a_r = 1, \\ -2, & if\ a_r = -1, \\ a_r + 1, & if\ a_q > a_r, \\ a_r - 1, & if\ a_q < a_r, \end{cases} \tag{21}$$

5. Encoder and decoder

The encoder of the proposed steganographic method based on modified BCH data hiding scheme and inserting-removing strategy is organized as follows:

For a given bitmap image I_m, payload P, quality factor Q_f, and secret key K process follows:

1. Divide image I_m into nonoverlapped 8 × 8 blocks of pixels and process DCT, quantization and rounding as presented in (16). Remove DC coefficients. Obtain a', a_q, a_r, and streams of DCT coefficients a. Permute stream a using K and any pseudo-random generator. Obtain stream $c = a \in (-\infty; -0.25) \cup (0.25; \infty)$ from the permuted stream a.
2. Define sets: modifiable c_m, insertable c_{Ins}, and removable c_R.
3. Define parameters for schemes 1 and 2, and number of the blocks k_1 and k_2 using (14) and (15). Divide message M into two parts: $M_1 = m_1^p \cdot k_1$ and $M_2 = m_2^p \cdot k_2$.
4. Start from the first block i = 1. Define the ith block of the DCT coefficients $c_{b_i} = c'_{m_i} \cup c'_{R_i} \cup c'_{Ins_i}$, where c'_{m_i}, c'_{R_i}, and c'_{Ins_i} are the modifiable, removable, and insertable subsets for the current block. If i = k_1 +1 switch to the scheme 2.
5. Define the block of nonzero rounded DCT coefficients $\mathrm{C}'_\mathrm{i} = c'_{m_i} \cup c'_{R_i}$.
6. Get the solutions for the block C'_i using the modified BCH data hiding scheme (see the algorithm in Section 3). Compute the distortion D for each solution using Equation (20). Choose solution J_m with the lowest distortion D_m and store it.
7. Modify the block C'_i by inserting or removing coefficients from the subsets c'_{R_i}, and c'_{Ins_i}. Obtain a new block: (i) after removing $C'_i = c'_{m_i} \cup c''_{R_i}$, where $c''_{R_i} = c'_{R_i} - c'_{R_i}(p)$ is the modified removable set and $c'_{R_i}(p)$ is the removed coefficient; (ii) after inserting $C'_i = c'_{m_i} \cup c'_{R_i} \cup c'_{Ins_i}(q)$, where $c'_{Ins_i}(q) = \pm 1$ is the inserted coefficient. p and q are the current position for insertion and removing.
8. Repeat steps 5-6 for all insertable and removable coefficients from c'_{R_i}, and c'_{Ins_i}.
9. Among all stored solutions J_m choose solution with the lowest distortion D_m. Modify one, two, or three coefficients according to the best solution (see explanation in Section 2) and, if necessary, insert or remove coefficient in the block c_{b_i}.
10. Process all k_1 + k_2 blocks using steps 4-9. Obtain the modified stream $c' = \{c_{b_1}, c_{b_2}, \ldots, c_{b_{k_2+k_2}}\}$.
11. Recover the original sequence order of the DCT coefficients a from the modified stream c' using the secret key K and utilized pseudo-random generator. Add DC coefficients, round the coefficients a', and obtain the modified JPEG image I'_m.

The decoder of the proposed steganographic method is organized as follows:

For the given modified JPEG image I'_m, quality factor Q_f, secret key K, and size of the payload $p = |P|$ process follows:

1. Read the DCT coefficients from the JPEG file. Permute them using the secret key K and utilized pseudo-random generator. Remove the DC coefficients. Obtain the stream of nonzero DCT coefficients C.
2. Using Equations (15) and (16) define parameters of the schemes 1 and 2, and the number of blocks k_1 and k_2. Here, $N = |C|$.
3. Divide C into the blocks according to the k_1 and k_2.
4. Decode data from each block using (9).

The steganographic method based only on modified BCH data hiding scheme skips the steps 7 and 8.

6. Experimental results

In these experiments, we try to hide different amount of data into the set of uncompressed images using the proposed BCH-based data hiding scheme with and without the inserting-removing strategy. The set of modified and original compressed images is analyzed by two powerful steganalysis algorithm proposed by Pevny and Fridrich

[20] and Kodovsky and Fridrich [25]. Those methods use 274 and 548 different features of the DCT coefficients, respectively. The union of the 274 or 548 features from the unmodified and modified images are used for making the models for the support vector machine (SVM) with parameter C = 10^4 and kernel width $\gamma = 10^{-4}$. A set of 4,000 natural uncompressed images (768*512) downloaded from Corel Draw and obtained from several digital cameras is used in our experiments. Proposed method needs 1-5 min for hiding data to each image. Experiments are carried out for seven different payloads (0.05, 0.1, 0.15, 0.17, 0.20, 0.22, and 0.25 bits per nonzero coefficient–bpc) and quality factor 75. SVM training process needs a set of 3,000 images (1,500 original and 1,500 stego images) for 7 different payload sizes. The SVM engine tests for 7 obtained models to test a set of 1,000 images (500 original and 500 stego) for 7 different payload sizes. The result shows the error probabilities of the steganalysis for each tested payload (see Figures 2 and 3).

The error probability is computed as follows:

$$e = \frac{1}{2}(P_a + P_b), \tag{28}$$

where P_a is the probability of misdetection (i.e., the unmodified image is classified as modified) and P_b is the probability of misclassification (i.e., the modified image is classified as unmodified).

In our experiments, we test both methods: (1) based only on the modified BCH-based data hiding scheme; and (2) the modified BCH-based data hiding scheme with the proposed inserting-removing strategy. The proposed methods achieve high error probability for all the tested payloads. For payloads up to 0.1 bpc, both methods have detectability close to 50%, meaning that the steganalysis cannot distinguish the unmodified images from the modified. This probability is almost equal to that of the coin toss. For higher payloads around 0.15 and 0.2 bpc, the proposed methods show much better performance compared to the MME. Significant improvement over the MME is justified on the fact of using methods with larger embedding efficiency (i.e., the BCH-based schemes with large m). The proposed method also shows better results compared to the methods based on the original BCH-based schemes. Hence, the proposed method with the inserting-removing strategy shows the significant improvement over the method with modified BCH-based data hiding scheme only, by 0.0363, 0.0414, and 0.0392 points in terms of error probabilities for payloads 0.15, 0.2, and 0.25, respectively. For payload of 0.25 bpc, both methods show 0.2961 and 0.3353 of the error probability. The error probabilities are better than those of the MME [14], original BCH-based [16], heuristic BCH-based scheme [17], and syndrome trellis code STC [22] proposed by Kodovsky and Fridrich. Such improvement was achieved by using modified BCH-based data hiding and unique inserting-removing strategy.

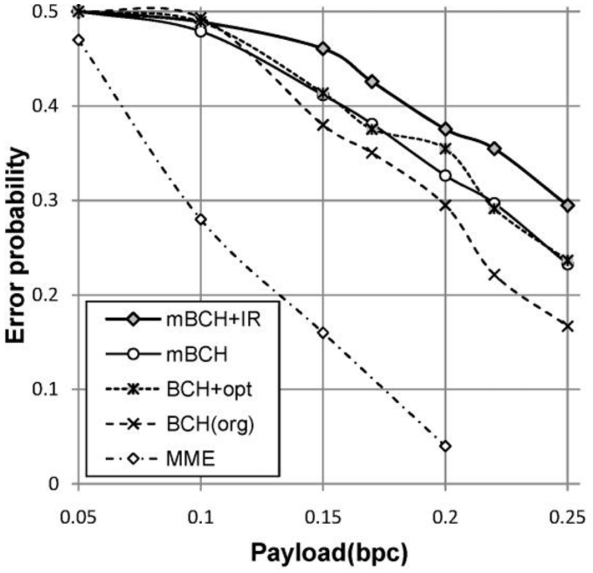

Figure 2 Error probability versus payload (bpc) for quality factor 75 using steganalysis [20].

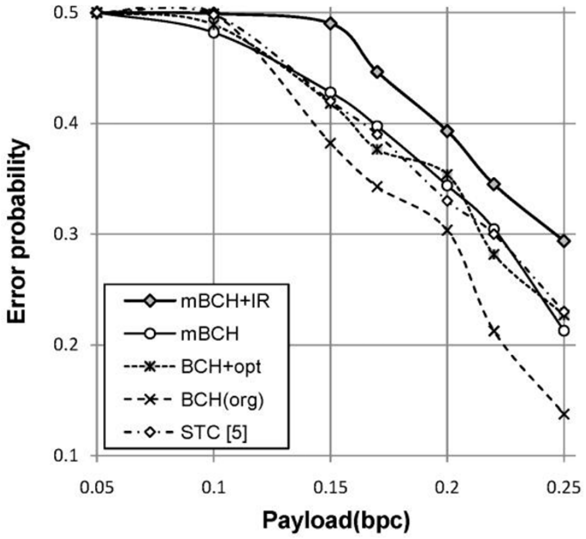

Figure 3 Error probability versus payload (bpc) for quality factor 75 using steganalysis [25].

7. Conclusion

In this article, an efficient data hiding technique for steganography is presented. The proposed BCH-based data hiding scheme uses two blocks to form a single combined block. A new data hiding strategy enables to get a joint solution for two blocks with intersected coefficients. Due to intersection, the proposed method requires small number of coefficients for hiding the same amount of data compared with the original nonoverlapping blockwise approaches. As a result, the proposed method can use the BCH-based schemes with large m (i.e., lager capacity). Even though the proposed method requires to use the same BCH-based scheme (for 0.17 and 0.2 bpc), the efficiency of data hiding is still high because the proposed two-scheme embedding has a lower ratio $k_1\backslash k_2$ compared to the original BCH-based scheme. The proposed BCH-based data hiding scheme significantly outperforms the MME and original BCH-based steganography in terms of the error probabilities and accuracy against the steganalysis. The proposed two-scheme embedding technique (see Equations 14 and 15) enables to use almost all the available DCT coefficients. The proposed strategy based on inserting and removing coefficients 1 or -1 increases the number of possible solutions and significantly decreases the total distortion. The experimental results show that the inserting-removing strategy significantly improves the performance of the proposed method. The combination of the modified BCH-based and the inserting-removing strategy achieves higher error probabilities and lower accuracy against the powerful steganalysis.

Acknowledgements
This study was supported by the Catholic University of Korea, National Research Foundation of Korea (grant 2011-0013695), ITRC and BK21 Project, Korea University and IT R&D program (Development of anonymity-based u-knowledge security technology, 2007-S001-01).

Author details
[1]School of Information, Communications, and Electronic Engineering, The Catholic University of Korea, Bucheon 420-743, Republic of Korea [2]CIST, Korea University, Seoul 136-701, Republic of Korea

Competing interests
The authors declare that they have no competing interests.

Received: 14 June 2011 Accepted: 26 April 2012
Published: 26 April 2012

References

1. N Provos, Defending against statistical steganalysis, in *Proc of 10th USENIX Security Symposium*, Washington, DC, 24–24 (2001)
2. J Eggers, R Bauml, B Girod, A communications approach to steganography, in *Proc of EI SPIE*, San Jose, CA, **4675**, 26–37 (2002)
3. H Noda, M Niimi, E Kawaguchi, Application of QIM with dead zone for histogram preserving JPEG steganography, in *Proc of ICIP*, Geneva, Italy, (2005)
4. K Solanki, A Sakar, BS Manjunath, YASS: Yet another steganographic scheme that resists blind steganalysis. Lect Notes Comput Sci. **2939**, 154–167 (2007)
5. A Westfeld, High capacity despite better steganalysis (F5–a steganographic algorithm). Lect Notes Comput Sci. **2137**, 289–302 (2001)
6. J Fridrich, Minimizing the embedding impact in steganography, in *Proc of ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2–10 (September 26–27, 2006)
7. J Fridrich, Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. Lect Notes Comput Sci. **3200**, 67–81 (2005)
8. J Fridrich, T Filler, Practical methods for minimizing embedding impact in steganography, in *Proc EI SPIE*, San Jose, CA, **6505**, 2–3 (2007)
9. J Fridrich, M Goljan, D Soukal, Perturbed quantization steganography using wet paper codes, in *Proc of ACM Workshop on Multimedia and Security*, Magdeburg, Germany, 4–15 (September 20–21, 2004)
10. J Fridrich, M Goljan, D Soukal, Perturbed quantization steganography. ACM Multimedia Secur J. **11**(2), 98–107 (2005)
11. J Fridrich, T Pevny, J Kodovsky, Statistically undetectable JPEG steganography: dead ends, challenges, and opportunities, in *Proc of ACM Workshop on Multimedia and Security*, Dallas, TX, 3–15 (September 20–21, 2007)
12. J Fridrich, M Goljan, D Soukal, Perturbed quantization steganography. ACM Multimedia Secur J. **11**(2), 98–107 (2005)
13. J Fridrich, M Goljan, D Soukal, Wet paper coding with improved embedding efficiency. IEEE Trans Inf Secur Forensics. **1**(1), 102–110 (2005)
14. YH Kim, Z Duric, D Richards, Modified matrix encoding technique for minimal distortion steganography. Lect Notes Comput Sci. **4437**, 314–327 (2006)
15. D Schönfeld, A Winkler, Reducing the complexity of syndrome coding for embedding. Lect Notes Comput Sci. **4567**, 145–158 (2008)
16. R Zhang, V Sachnev, HJ Kim, Fast BCH syndrome coding for steganography. Lect Notes Comput Sci. **5806**, 48–58 (2009)
17. V Sachnev, HJ Kim, R Zhang, Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding, in *Proc of ACM Workshop on Multimedia and Security*, Princeton, NJ, 131–139 (September 7–8, 2009)
18. T Filler, J Fridrich, Steganography using Gibbs random fields, in *Proceedings of ACM Multimedia and Security Workshop*, Rome, Italy, 199–212 (2010)
19. D Upham, http://www.funet.fi/pub/crypt/stegangraphy/jpeg-jsteg-v4.diff.gz
20. T Pevny, J Fridrich, Merging Markov and DCT features for multi-class JPEG steganalysis, in *Proc of SPIE*, San Jose, CA, **6505**, 3–4 (2007)
21. YQ Shi, C Chen, W Chen, Markov process based approach to effective attacking JPEG steganography. Lect Notes Comput Sci. **4437**, 249–264 (2006)
22. T Filler, J Judas, J Fridrich, Minimizing embedding impact in steganography using trellis-coded quantization. IEEE Trans Inf Secur Forensics. **6**(3), 920–935 (2011)
23. H Rifa-Pous, J Rifa, Product perfect codes and steganography. Digital Signal Process. **19**, 764–769 (2009)
24. Z Zhao, F Wu, S Yu, J Zhou, A lookup table based fast algorithm for finding roots of quadratic or cubic polynomials in the GF(2^m). J Huazhong Univ Sci Technol (Nat Sci Ed.). **33**(1), 70–73 (2005)
25. J Kodovsky, J Fridrich, Calibration revisited, in *Proceedings of the 11th ACM Multimedia & Security Workshop*, ed. by J, Dittmann, S, Craver, J, Fridrich Princeton, NJ, (Septmber 7–8, 2009)

doi:10.1186/1687-6180-2012-89
Cite this article as: Sachnev and Kim: **Modified BCH data hiding scheme for JPEG steganography.** *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:89.