A Novel Turbo Unequal Error Protection Scheme for Image Steganography

Qian MAO*, Chuan QIN
School of Optical-Electrical and Computer Engineering
University of Shanghai for Science and Technology
No. 516, Jungong Rd., Yangpu, Shanghai 200093, P.R. China
e-mail: maoqiansh@gmail.com, qin@usst.edu.cn

Received: December 2011; accepted: August 2013

Abstract. An Unequal Error Protection (UEP) scheme for image steganography is proposed in this paper. When stego images are transmitted through a noisy channel, the embedded message may be more sensitive to noise than the cover message. Therefore, we propose a Turbo UEP coding scheme for steganographic communication in a noisy channel, which provides higher error protection for the embedded message and lower error protection for the cover message. Simulations show that this coding scheme provides different error protection levels in one coding process and maintains the coding rate constant. In addition, an application scheme of the proposed Turbo UEP code for steganographic communication is presented, and the experimental results show that the extracted secret images have better quality than the cover images after decoding.

Key words: image steganography, unequal error protection, Turbo code, puncture.

1. Introduction

Steganography is a popular information hiding technology. It transmits secret messages by embedding them imperceptibly in data, images, audio, and video, which are known as cover messages. Legal receivers acquire the secret message by using an extraction algorithm, while the illegal attackers cannot detect the secret message. The focus of steganography is to embed as much information as possible while modifying the cover message as little as possible. Among all these carrier objects, images are the most frequently used. Using the redundancy of the images, digital image steganography can embed extensive secret messages imperceptibly and transmit them together with the cover images (Fridrich and Lisoněk, 2007).

The key feature of steganography is the imperceptibility of the embedded message. In digital image steganography, although only the most insignificant components in the cover image are altered, many analytical techniques can reveal the existence of the embedded message by statistical methods. In order to resist such steganalysis, the following two measures may be taken: (a) avoid conspicuous parts when embedding meassages in the cover

*Corresponding author.

and (b) improve the embedding effciency, i.e., embed more information per modification to the cover image (Zhang *et al.*, 2007).

The first goal can be achieved by various means, including wet paper codes and adaptive steganography. In the wet paper codes, we imagine that the cover image was exposed to rain and that the sender can modify the dry spots in the cover image slightly but cannot modify the wet spots. The "rain" could be random, pseudo-random, or another format, which is determined exclusively by the sender. The recipient does not have to determine the dry pixels in the stego image in order to read the secret message (Fridrich *et al.*, 2006). One mechanism for determining the dry pixels is adaptive steganography. In this method, the "rain" is not random; rather, it is determined by the characteristics of the cover image and the size of the secret message. For example, the smooth regions in the cover image will be embedded with less secret message, since it is more sensitive for data hiding (Luo *et al.*, 2010). Both wet paper codes and adaptive steganograhphy provide improved steganographic security and are less vulnerable to steganalytic attacks.

The second method for resisting steganalysis is to modify as few pixels as possible in the cover image, while embedding the same amount of secret message. This can be achieved by some coding mechanisms, such as matrix embedding (Fridrich and Soukal, 2006) and dynamical running coding. In matrix embedding steganography, the cover coefficients are perturbed minimally, such that the transmitted bits fall in a coset of the linear code, with the syndrome conveying the hidden bits (Sarkar *et al.*, 2010). Dynamical running coding steganography works in a running manner. In this way, each secret bit is represented by a series of consecutive cover bits, and each available cover bit also relates to several consecutive secret bits (Zhang and Wang, 2006). These kinds of stego methods increase the embedding effciency at the cost of decreasing the embedding rate, and enhance the security of the steganography.

When stego images are transmitted in a noisy channel, they will be affected by the channel noise. In order to resist the errors that occur in the transmission channel, error correction codes are necessary. Traditional steganopraphic communication implements the error correction coding process after embedding the secret message, which means the error correction code provides the same error correction capabilities for both the cover image and the secret message. However, for many steganographic methods, the embedded message is less robust to the channel noise than the cover message. Taking the second steganographic class mentioned above, for example, both the matrix embedding and the dynamical running coding take several cover bits and secret bits as a coding group and embed the secret bits by coding them together. Any error that occurs in the codeword may cause the collapse of decoding and destroys all of the secret bits in this coding group. Therefore, steganography sometimes makes the embedded message more sensitive to channel noise. This means higher error correction capabilities are necessary for the embedded massage.

In order to provide different error correction capabilities for the embedded bits and the cover image bits, another, more sophisticated coding scheme was proposed. In this scheme, there are two error correction coding procedures in a steganographic communication system, i.e., (1) the secret message is encoded by an error correction coding scheme first and (2) the entire stego image is encoded by another error correction coding scheme after that. The two error correction schemes may have different coding rates and error correction capabilities, and they are often processed independently (Sarkar *et al.*, 2010). It is apparent that the two-step coding mechanism provides higher error protection for the secret message but has higher computational complexity.

In this paper, we propose a novel error correction coding scheme for steganographic communication that is more efficient. In this scheme, we combine the two error correction coding processes into one, but we still provide different error correction capabilities for the embedded bits and the cover bits, according to their sensitivities to noise. This can be realized by the Unequal Error Protection (UEP) codes, which are a kind of error correction coding scheme. The UEP codes have two notable advantages, i.e., (1) they provide different error correction capabilities for information bits in one coding process and (2) they maintain a constant coding rate. Since UEP codes provide different error correction capabilities but do not increase the computational complexity and bandwidth, they are suitable for the unequal error protection communication.

The UEP codes were first proposed by Masnick and Wolf (1967). In this scheme, each digit of the codeword is assigned an error protection level f_i. Then, if f errors occur in the reception of a codeword, all digits that have protection f_i greater than or equal to f will be decoded correctly, even though the entire codeword may not be decoded correctly. Masnick's UEP code is a kind of linear block code, and, after it was proposed, extensive research has been done in this field. Gils proposed a cyclic UEP code and proved its UEP capability (Gils, 1983). In this scheme, the generator matrix of the cyclic code was improved to make the code space provide better error correction capability for the important information bits. After that, Lin found all the cyclic UEP codes whose lengths were less than 65 by computer searching (Lin *et al.*, 1990). But neither of them proposed an effective decoding algorithm for the cyclic UEP codes.

Most of the existing UEP codes are linear. In recent years, Turbo codes have gained a lot of attention and have been used in many applications because of their perfect error correction performance (Berrou and Glavieux, 1996; Benedetto and Montorsi, 1996; Chatzigeorgiou *et al.*, 2009). Berrou *et al.* (1993) were the first to propose Turbo codes that have excellent error-correcting performance. If the block length is large enough, the Bit Error Rate (BER) performance of the Turbo code will be very near to the Shannon limitation. For example, when the Signal-to-Noise Ratio (SNR) of the Additive White Gaussian Noise (AWGN) channel is 0.7 dB and the Binary Phase Shift Keying (BPSK) modulation is used, the $1/2$ coding rate Turbo code obtains a 10-5 BER. In many cases, a Turbo encoder generates two parity bits for every input bit, therefore, the coding rate without puncturing is $1/3$. In order to obtain a higher coding rate, a puncturing mechanism is often used (Haccoun and Bégin, 1989; Hagenauer, 1988). By periodically eliminating some bits from the output of the Recursive Systematic Convolutional (RSC) encoders of the Turbo encoder, a higher coding rate can be achieved. The performance of the punctured Turbo codes has been widely researched (Chatzigeorgiou *et al.*, 2009; Chatzigeorgiou *et al.*, 2006; Kousa and Mugaibel , 2002). Most puncturing mechanisms delete the parity bits periodically and uniformly in a coding block. Barbulescu proposed a Turbo UEP code with two error protection levels based on a rate-compatible puncture mechanism (Barbulescu and Pietrobon, 1995). In this scheme, the parity bits are deleted non-uniformly in a coding block, and the information bits that have more parity bits are provided higher error correction capability. This scheme causes various block lengths after encoding. In order to overcome this issue, a special modulation or interleaving scheme must be used. After that, a lot of work has been done to study and improve the UEP capabilities of the Turbo codes (Aydinlik and Salehi, 2009; Henkel and Deetzen, 2006; Zhou and Xu, 2005).

This paper improves Barbulescu's scheme and proposes a new Turbo UEP code that keeps the coding rate constant. After that, the proposed Turbo UEP code was applied in image steganograghy and provided improved error protection for the embedded message. Section 2, which follows, gives some notations and a brief introduction of the decoding algorithm, and Section 3 describes the proposed Turbo UEP code. Section 4 presents the application scheme of the Turbo UEP code in steganographic image communication, and Section 5 presents our conclusions.

2. Preliminaries

2.1. *Notations*

In this section, we give some notations for Turbo encoding and decoding, which are shown in Table 1.

2.2. *Log-Maximum a Posteriori (log-MAP) Algorithm*

In our Turbo UEP scheme, the *log-maximum a posteriori* (log-MAP) algorithm is applied to the RSC code on a AWGN channel of which energy per bit to noise power spectral density ratio is E_s/N_0. The log-MAP algorithm judges the decoding output by the *Log-Likelihood Ratio* (LLR) (Bahl *et al.*, 1974). Assuming that the channel is binary and the

Table 1
Some notations for Turbo codes.

Symbol	Definition
L	Block length of Turbo code
L_1	Length of high-protection block
L_2	Length of low-protection block
d	The original information bit
r	The received information bit
$\hat{d}$	The output bit of the decoder after judgment
K	The constraint length of the RSC encoder
Y	The parity bit
R	Coding rate
N	Number of pixels in a Turbo block

block length of Turbo code is L, the LLR value of the information bit, d_k $(k = 1, \ldots, L)$, is:

$$L(d_k) = \ln\left[\frac{P(d_k = 1|r_k)}{P(d_k = 0|r_k)}\right], \tag{1}$$

where $P(d_k = i|r_k)$ $(i = 0, 1)$ is the conditional probability of $d_k = i$ under the condition of the received digit r_k. The parameter $P(d_k = i|r_k)$ is defined as the *a posteriori probability* (APP) value of r_k. By incorporating the RSC code trellis into the computation of $P(d_k = i|r_k)$, the APP value can be denoted as the following form:

$$P(d_k = i|r_k) = \frac{P(d_k = i, r_k)}{P(r_k)} = \frac{\sum_{(s',s)\in\Sigma_k^i} p(s_k = s', s_{k+1} = s, r_k)}{P(r_k)}, \quad i = 0, 1, \tag{2}$$

where s and s' are the states of the RSC code trellis and Σ_k^i denotes the set of all the two consequent states, s_k and s_{k+1}, of which input bit, d_k, is i. Therefore, Eq. (1) can be rewritten as:

$$L(d_k) = \ln\left[\frac{P(d_k = 1|r_k)}{P(d_k = 0|r_k)}\right] = \ln\frac{\sum_{(s',s)\in\Sigma_k^1} p(s_k = s', s_{k+1} = s, r_k)}{\sum_{(s',s)\in\Sigma_k^0} p(s_k = s', s_{k+1} = s, r_k)}. \tag{3}$$

Using the log-MAP algorithm proposed by Bahl *et al.* (1974), the probability of $p(s_k = s', s_{k+1} = s, r_k)$ can be presented as the following form:

$$p\big(s_k = s', s_{k+1} = s, r_k\big) = e^{\alpha_k^*(s') + \gamma_k^*(s',s) + \beta_{k+1}^*(s)} \tag{4}$$

where $\alpha_k^*(s')$, $\gamma_k^*(s', s)$, and $\beta_{k+1}^*(s)$ are the forward metric, backward metric, and branch metric of the RSC decoder, and

$$\alpha_{k+1}^*(s) = \max\nolimits_{s'}^*\big[\gamma_k^*(s', s) + \alpha_k^*(s')\big], \tag{5}$$

$$\beta_k^*(s') = \max\nolimits_s^*\big[\gamma_k^*(s', s) + \beta_{k+1}^*(s)\big], \tag{6}$$

$$\gamma_k^*(s', s) = \frac{d_k L_a(d_k)}{2} + \frac{L_c}{2} r_k d_k. \tag{7}$$

In the above functions, parameter $L_c = 4E_s/N_0$ is the channel reliability factor, $L_a(d_k)$ is the *a priori* LLR value of d_k, which is equal to the extrinsic LLR value, $L_e(d_k)$, of the other decoder except the first iteration (for the first iteration, $L_a(d_k) = 0$ for the equiprobable bits), and function $\max^*(x, y)$ is

$$\max{}^*(x, y) \equiv \ln\big(e^x + e^y\big). \tag{8}$$

According to the trellis diagram of the RSC code and (3)–(8), function (1) can be deduced to the following form:

$$L(d_k) = L_c r_k + L_a(d_k) + L_e(d_k) \tag{9}$$

where $L_e(d_k)$ is the extrinsic LLR value of d_k. The LLR value, $L(d_k)$, can be obtained by the RSC code trellis. Thus, the extrinsic LLR value can be obtained, i.e., $L_e(d_k) = L(d_k) - L_c r_k - L_a(d_k)$. The extrinsic LLR value, $L_e(d_k)$, will be transmitted to the other decoder as the *a priori* LLR value.

From above functions we see that iterative computations are implemented between the two decoders, i.e., the extrinsic LLR value output from one decoder is transmitted to the other decoder as the *a priori* LLR value. Each iteration makes the LLR value, $L(d_k)$, closer to the original information bit. After several times of iteration, the Turbo decoder judges the result $\hat{d}_k$ according to $L(d_k)$, i.e.,

$$\hat{d}_k = \begin{cases} 1 & \text{if } L(d_k) > 0, \\ 0 & \text{if } L(d_k) < 0. \end{cases} \tag{10}$$

3. Proposed Turbo Unequal Error Protection Codes

In this section, we propose an advanced, rate-compatible, Turbo UEP code. In our scheme, two error protection levels are provided, and the coding rate of the entire block is $1/2$.

3.1. *Proposed Turbo UEP Encoder*

The structure of our Turbo UEP encoder is shown in Fig. 1. The encoder consists of one interleaver and two RSC encoders. In order to obtain the UEP capabilities, an advanced, rate-compatible, puncturing mechanism is used in the outputs of the two RSC encoders.

Assuming that the constraint length of the RSC encoder is K, the memory is $M = K - 1$, to achieve high error correction performances of Turbo code, the RSC encoders should satisfy the following requirements: (1) A short constraint length usually provides better error correction performance in medium BER levels, therefore, in most cases, $K \leqslant 4$; (2) the two RSC encoders are usually the same, while, two different RSC encoders are also used in some Turbo encoders and they can also provide good performances; and (3) the performances of RSC encoders are much better than that of the Feed-forward Convolutional encoders in a Turbo encoder.

Therefore, in the proposed Turbo UEP coding scheme, the two RSC encoders are the same, the coding rate of the RSC encoders is $1/2$, and the generator matrix is

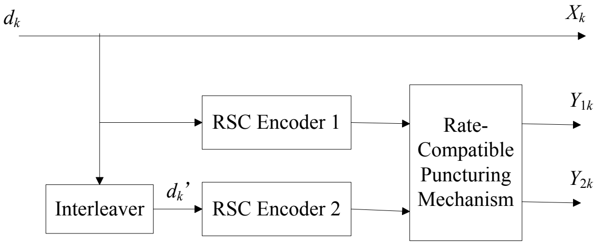


Fig. 1. Structure of the Turbo UEP encoder.

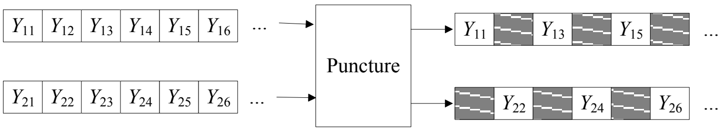

Y_{11} Y_{12} Y_{13} Y_{14} Y_{15} Y_{16} ...
Y_{21} Y_{22} Y_{23} Y_{24} Y_{25} Y_{26} ...
Puncture
Y_{11} Y_{13} Y_{15} ...
Y_{22} Y_{24} Y_{26} ...

Fig. 2. The puncturing mechanism of the Turbo EEP codes.

$G = [1\ G_2/G_1]$, where $G_1 = [g_{10}, g_{11}, \ldots, g_{1,K-1}]$ and $G_2 = [g_{20}, g_{21}, \ldots, g_{2,K-1}]$. Then, for the kth information bit d_k, there are two outputs of the RSC encoder, as shown in the following:

$$X_k = d_k, \tag{11}$$

$$Y_k = \sum_{i=0}^{K-1} g_{2i} a_{k-i} \mod 2, \tag{12}$$

where

$$a_r = d_r + \sum_{i=1}^{K-1} g_{1i} a_{r-i} \mod 2, \quad r = k, k-1, \ldots, k-(K-1). \tag{13}$$

In the above functions, X_k is defined as systematic output and Y_k is the parity bit of d_k. For the Turbo encoder, there are three outputs of the kth input bit d_k, as shown in Fig. 1. The first output, X_k, is equal to d_k. The second output, Y_{1k}, is the parity bit of d_k computed by RSC encoder1 using (12) and (13), i.e., $Y_{1k} = Y_k|_{d_k}$. For the third output, the original bit sequence is randomly permuted by an interleaver first, after that, the permuted information bits are input into the RSC encoder2, which has the same structure as RSC encoder1. Assuming that the kth information bit for the third output is d'_k, the output is $Y_{2k} = Y_k|_{d'_k}$, which is obtained by (12) and (13) with the input of d'_k.

The Turbo encoder outputs three bits for every input bit. Therefore, the coding rate without puncturing is $1/3$. In order to increase the coding rate, a puncturing mechanism is often used. In the puncturing mechanism of the Turbo EEP code, the deleted bits are usually located periodically, as shown in Fig. 2. In Fig. 2, Y_{1i} $(i = 1, 2, \ldots)$ denotes the ith output bit of RSC encoder1 in Fig. 1, and Y_{2i} $(i = 1, 2, \ldots)$ denotes the ith output bit of RSC encoder2. The puncturing algorithm deletes the bits on the even locations in Y_{1i} and the bits on the odd locations in Y_{2i}. By this means, the parity bits are reduced by half, and the coding rate of the Turbo code is increased from $1/3$ to $1/2$.

After puncturing, the transmission sequence is

$$X_1, Y_{11}, X_2, Y_{22}, X_3, Y_{13}, X_4, Y_{24}, X_5, Y_{15}, X_6, Y_{26}, \ldots, \tag{14}$$

where X_i $(i = 1, 2, \ldots)$ is the ith bit of the first output of the encoder, as shown in Fig. 1.

X_1	X_2	……	X_{L_1}	X_{L_1+1}	X_{L_1+2}	X_{L_1+3}	X_{L_1+4}	X_{L_1+5}	X_{L_1+6}	X_{L_1+7}	X_{L_1+8}	……	$X_{L\mathrm{Turbo}}$
Y_{11}	Y_{12}	……	Y_{1L_1}	$Y_{1(L_1+1)}$	▨	$Y_{1(L_1+3)}$	▨	$Y_{1(L_1+5)}$	▨	▨	▨	……	▨
Y_{21}	Y_{22}	……	Y_{2L_1}	▨	$Y_{2(L_1+2)}$	▨	$Y_{2(L_1+4)}$	▨	$Y_{2(L_1+6)}$	▨	$Y_{2(L_1+8)}$	……	$Y_{2(L\mathrm{Turbo})}$

Fig. 3. The puncturing mechanism of the Turbo EEP codes.

It is clear that only parity bits can be punctured, since deletion of the systematic bits leads to inferior performance for decoding. If one parity bit is reserved for every t information bits, the coding rate R is

$$R = \frac{t}{t+1}. \tag{15}$$

Different puncturing mechanisms result in different error protection levels. The more the parity bits are reserved, the higher the error protection level that is provided for the systematic bit. Therefore, we improve the puncturing mechanism to obtain the UEP capability. In our Turbo UEP scheme, we partition all the L information bits in a coding block into two parts with different error protection levels, the lengths of which are L_1 and L_2, respectively. It is obvious that $L_1 + L_2 = L$. We denote this code as an (L, L_1) Turbo UEP code. The first L_1 information bits are provided a higher error protection level, and the remaining $L2$ information bits are provided a lower error protection level. In order to obtain this Unequal Error Protection capability, we used the following advanced, rate-compatible, puncturing mechanism.

- For the first L_1 information bits, both Y_{1i} and Y_{2i} are reserved when puncturing. So the coding rate for these L_1 information bits is $1/3$.
- For most of the remaining L_2 information bits, Y_{1i} and Y_{2i} are punctured periodically, i.e., we delete Y_{1i} if i is even and delete Y_{2i} if i is odd. But in order to obtain a $1/2$ coding rate for the entire block, we choose L_1 information bits among these L_2 information bits randomly and delete both Y_{1i} and Y_{2i} of them.

This puncturing scheme is shown in Fig. 3. In this example, the $(L_1 + 7)$th information bit in the low error protection region is selected randomly, and both of its parity bits are deleted.

3.2. *Proposed Turbo UEP Decoder*

At the receiver side, the decoder uses the same puncturing algorithm to classify X_i, Y_{1i}, and Y_{2i} in the received sequence, sends them to a Turbo decoder, and starts an iterative decoding process. The structure of the Turbo UEP decoder is shown in Fig. 4.

The log-MAP algorithm is used in the decoding process. Therefore, the LLR value, $L(d_k)$, shown as (9), is computed in each constituent decoder. This soft output is divided into three parts, i.e., (1) a weighted version of the systematic input $L_c r_k$, which is a part of the input of the decoder; (2) a copy of the input *a priori* information $L_a(d_k)$, which is equal to zero for the first iteration under the condition that the input is equiprobable and is

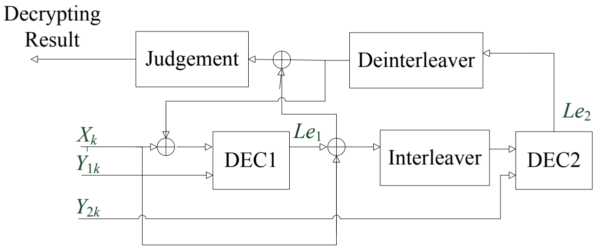


Fig. 4. Structure of the Turbo UEP decoder.

the extrinsic LLR value $L_e(d_k)$ of the other decoder for the subsequent iterations; and (3) the extrinsic LLR value $L_e(d_k)$, which is derived from the current stage of decoding and will be transmitted to the other decoder as its *a priori* LLR value. After several iterations, the Turbo decoder judges the result according to (10).

3.3. *BER Performance of the Proposed Turbo UEP Code*

In the following, we prove the error correction capabilities of the proposed Turbo UEP code by simulating BER versus SNR. The SNR is defined as:

$$\mathrm{SNR} = 10 \log_{10} \frac{E_b}{N_0} \text{ (dB)}. \tag{16}$$

Assuming that there are Q information bits input into the Turbo encoder and that q of them occur errors after transmission and decoding, the BER is calculated by the following equation:

$$\mathrm{BER} = \frac{q}{Q} \times 100\%. \tag{17}$$

As we know, the bit error probability, P_b, of the Binary Symmetric Channel (BSC) without error correction coding is:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \tag{18}$$

Therefore, the higher the SNR is, the lower the bit error probability is. The bit error probability shown as (18) is provided by a BSC, which can be constructed by an AWGN channel with binary modulation. Employing some error correcting codes, the BER of the decoded information bits can be decreased. In the following, the BER of our proposed Turbo UEP scheme is measured by simulation.

In the following simulation, the input data are numbers varying from 0 to 255, the transmission channel is the AWGN channel with binary modulation (BPSK) in optical communication, and the parameters of the Turbo code are listed in Table 2.

Table 2
Parameters of the Turbo code.

Generate matrix	$g = [111; 101]$
Decoding algorithm	Log-MAP
Iteration number	5
Block length	514 bits

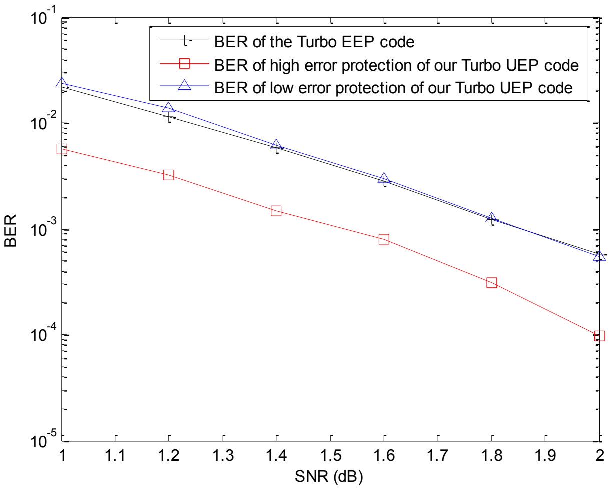


Fig. 5. BER performances of the Turbo EEP code and the proposed Turbo UEP code.

Figure 5 shows the BER performances of the Turbo EEP code and the proposed Turbo UEP code. Note that for any given SNR, the terminating condition of the simulation program is that 10 000 error bits occur after decoding. Therefore, the confidence level (CL) of BER measurements is $CL = 1 - e^{-Q*BER} = 1 - e^{-q} = 1 - e^{-10\,000} \approx 100\%$. In this simulation, the length of the high error protection level of the Turbo UEP code is 16 bits. From this figure, we find that the BER performance of the high error protection bits in our Turbo UEP code is better than that of the Turbo EEP code. This is because of the high coding rate of these bits. For the low error protection bits, although there are L_1 information bits without parity bits, they are less and located randomly. So the BER performance of the low error protection bits in our Turbo UEP code is almost as good as that of the Turbo EEP code, as shown in Fig. 5.

Table 3 shows the BER performances of the Turbo EEP code and three Turbo UEP codes with different length of high error protection level when the SNR is 2.0 dB. The length of coding block of all the schemes is 514 bits.

From this table, we find that the BER performances of the high error protection bits are always better than those of the Turbo EEP code in the three UEP schemes. And the greater the amount of high error protection information bits, the worse the average BER performance of the entire block became. This is because excessive high error protection bits cause many low error protection bits to lose their parity bits, which reduces the Turbo decoding performance.

Table 3
BER performances of a Turbo EEP code and three Turbo UEP codes.

	Turbo EEP code	Turbo UEP scheme I	Turbo UEP scheme II	Turbo UEP scheme III
Length of L_1	\	8 bits	16 bits	32 bits
BER of L_1	\	1.266e−4	9.707e−005	1.231e−4
Length of L_2	\	506 bits	498 bits	482 bits
BER of L_2	\	4.444e−4	5.463e−004	7.053e−4
Coding rate	1/2	1/2	1/2	1/2
Average BER	5.712e−004	4.394e−4	5.323e−004	6.689e−4

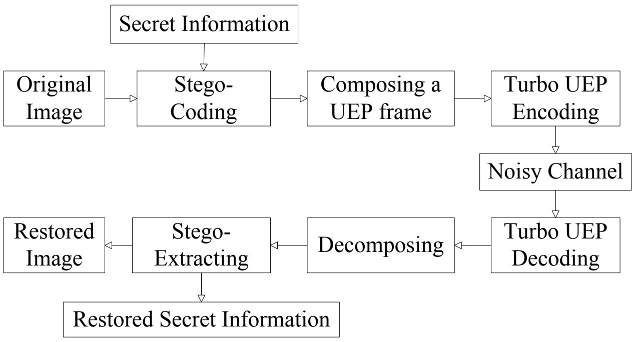


Fig. 6. Flowchart of the stego-communication using Turbo UEP code.

4. Applications of the Turbo UEP Codes in Image Steganography

For a digital image in the spatial domain, the change of the lowest bit of the pixel's gray value has the least influence on the image. Therefore, the secret message is usually embedded in these bits. Data embedding usually makes the secret-carry-bits more sensitive to noise. In order to provide a high error correction to the embedded information, an Unequal Error Protection steganographic communication system is proposed, which is shown in Fig. 6.

In order to provide higher error protection capability to the embedded information in our UEP steganographic communication system, the stego image is processed as follows:

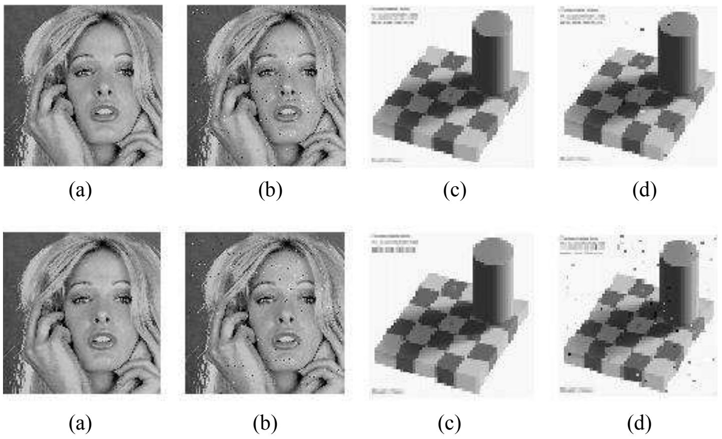
- Take N pixels from the stego-image to compose a coding block; then, the block length of the Turbo UEP code is

$$L = 8 * N + l, \tag{19}$$

  where l is the length of the tail bits of the Turbo encoder.
- Take all the secret bits in a coding block and put them in the first $L1$ locations; then, the remaining bits are in the last $(8N - L_1)$ locations.
- Encode this coding block by a (L, L_1) Turbo UEP encoder.

Table 4
BERs of the stego-communication with different embedding rates.

Cover image	Embedding rate	UEP scheme		EEP scheme
		Secret message	Cover image	
Tiffany	10%	1.5263e−004	3.6526e−004	3.2997e−004
	30%	1.2716e−004	5.703e−004	3.3855e−004
	45%	1.4412e−004	6.9809e−004	3.7575e−004
Lena	10%	3.2826e−005	4.5167e−004	3.5772e−004
	30%	6.564e−005	5.2469e−004	3.5008e−004
	45%	5.8346e−005	5.6038e−004	3.6552e−004
Baboon	10%	3.8156e−005	3.3522e−004	3.0613e−004
	30%	8.9013e−005	5.4932e−004	3.8576e−004
	45%	1.1021e−004	6.7425e−004	2.9945e−004



(a) (b) (c) (d)

(a) (b) (c) (d)

Fig. 7. Experimental results 1.

At the receiver side, a reverse process is implemented. Since the secret message is provided high error protection, the BER performance will be improved. Table 4 shows the BERs of the cover image and the secret message with different embedding rates when the SNR of the AWGN channel is 2.0 dB and the BPSK modulation is employed. In these experiments, the cover images are bmp images and the secret message is binary data that are generated randomly. From these experimental results, we see that the BERs of the secret message are better than that of the cover images. And with the increasing of the embedding rate, this superiority decreases. This is because that too much high error protection bits reduce the decoding performance of the Turbo UEP code, as shown in Section 3.3.

Figures 7–9 show some experimental results in which both the cover message and the embedded message are images. The sizes of the cover images and the embedded images are 512 × 512 and 64 × 64, respectively. The SNR of the AWGN channel is 1.5 dB. In Figs. 7–9, the first row shows the experimental results using the proposed Turbo UEP scheme, and the second row shows the results using the Turbo EEP scheme. And (a) is the

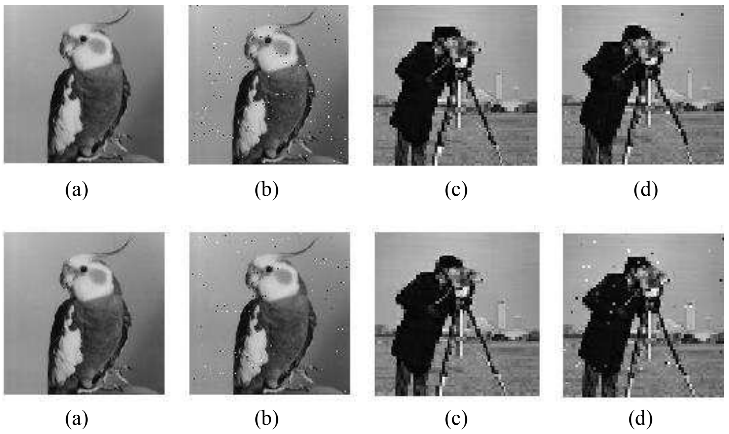


Fig. 8. Experimental results 2

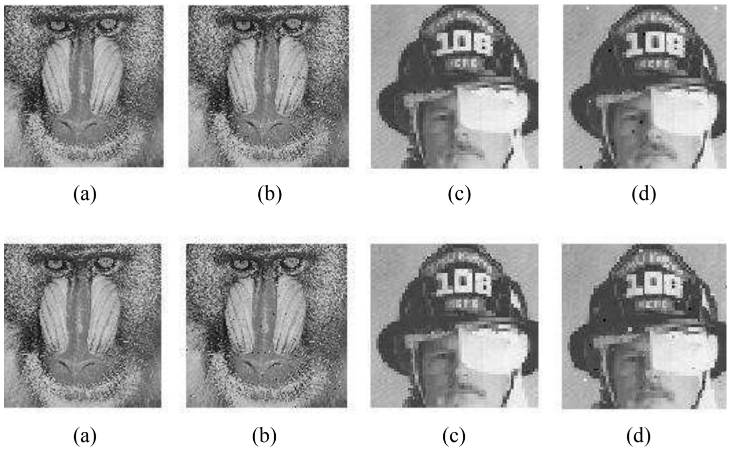


Fig. 9. Experimental results 3.

stego image of the sender, (b) is the restored stego image of the recipient after decoding, (c) is the original secret image, and (d) is the extracted secret image. Table 5 shows their Peak Signal to Noise Ratios (PSNRs), which is computed by the following function:

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{S_h S_v} \sum_{i=1}^{S_h} \sum_{j=1}^{S_v} \left[p_o(i, j) - p_s(i, j) \right]^2}, \tag{20}$$

where $S_h \times S_v$ is the size of the cover image, and $p_o(i, j)$ and $p_s(i, j)$ are pixels (i, j) in the original cover image and the stego image, respectively.

From the experiments, we found that, in our Turbo UEP scheme, the extracted secret images had better image quality than the stego images due to their high error protection level. While in the Turbo EEP scheme, both the secret images and the stego images have the same image quality.

Table 5
PSNR (dB) of teh stego images and the secret images using Turbo UEP code and Turbo EEP code.

Experment index	Turbo UEP scheme		Turbo EEP scheme	
	Stego image	Secret image	Stego image	Secret image
1	26.96	35.499	29.059	29.046
2	27.704	38.394	28.177	27.517
3	27.986	35.324	29.228	31.786

5. Conclusion

A novel Turbo UEP coding scheme with two error protection levels is proposed in this paper. In our UEP scheme, both of the two parity bits are reserved for the high error protection bits. Most of the low protection bits hold one parity bit, but in order to obtain a $1/2$ coding rate, some low error protection bits were selected randomly and do not have a parity bit. Simulations showed that our UEP scheme, compared with the Turbo EEP code, provided better BER performance for the high error protection bits and equivalent BER performance for the low error protection bits. After that, we applied our Turbo UEP code to steganographic image communication and provided better error protection to the embedded message, since it was more sensitive to noise and usually more important. Simulations showed that our scheme improved the BER performance of the secret message when transmitted in a noisy channel and the coding rate was constant.

References

Açkel, Ö.F., Ryan, W.E. (1999). Punctured Turbo-codes for BPSK/QPSK channels. *IEEE Transactions on Communications*, 47(9) 1315–1323.

Aydinlik, M., Salehi, M. (2009). Performance bounds for unequal error protecting Turbo codes. *IEEE Transactions on Communications*, 57(5), 1215–1220.

Barbulescu, A.S., Pietrobon, S.S. (1995). Rate compatible Turbo codes. *Electronics Letters*, 31(7), 535–536.

Berrou, C., Glavieux, A. (1996). Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on Communications*, 44(10), 1261–1271.

Berrou, C., Glavieux, A., Thitimajshima, P. (1993). Near Shannon limit error-correcting coding and decoding: turbo codes. In: *Proceedings of IEEE International Conference on Communication*, pp. 1064–1070.

Benedetto, S., Montorsi, G. (1996). Unveiling Turbo codes: some results on parallel concatenated coding schemes. *IEEE Transactions on Information Theory*, 42(2), 409–428.

Chatzigeorgiou, I., Rodrigues, M.R.D., Wassell, I.J., Carrasco, R. (2006). A novel technique for the evaluation of the transfer function of punctured Turbo codes. In: *Proceedings of IEEE International Conference on Communication*, pp. 1–6.

Chatzigeorgiou, I., Rodrigues, M.R.D., Wassell, I.J., Carrasco, R.A. (2009). Analysis and design of punctured rate-1/2 Turbo codes exhibiting low error floors. *IEEE Journal on Selected Areas in Communications*, 27(6), 944–953.

Fridrich, J., Lisoněk, P. (2007). Grid coloring in steganography. *IEEE Transactions on Information Theory*, 53(4), 1547–1549.

Fridrich, J., Soukal, D. (2006). Matrix embedding for large payloads. *IEEE Transactions on Information Forensics and Security*, 1(3), 390–395.

Fridrich, J., Goljan, M., Soukal, D. (2006). Wet paper codes with improved embedding efficiency. *IEEE Transactions on Information Forensics and Security*, 1(1), 102–110.

Gils, W.J.V. (1983). Two topics on linear Unequal Error Protection codes: bounds on their length and cyclic code classes. *IEEE Transactions on Information Theory*, IT-29(6), 866–876.

Haccoun, D., Bégin, G. (1989). High-rate punctured convolutional codes for Viterbi and sequential decoding. *IEEE Transactions on Communications*, 37(11), 1113–1125.

Hagenauer, J. (1988). Rate compatible punctured convolutional codes and their applications. *IEEE Transactions on Communications*, 36(4) 389–400.

Henkel, W., Deetzen, N.V. (2006). Path pruning for unequal error protection Turbo codes, In: *Proceedings of International Zurich Seminar on Communications*, pp. 142–145.

Kousa, M.A., Mugaibel, A.H. (2002). Puncturing effects on Turbo codes. *IEE Proceedings – Communications*, 149(3), 132–138.

Lin, M.C., Lin, C.C., Lin, S. (1990). Computer search for binary cyclic UEP codes of odd length up to 65. *IEEE Transactions on Information Theory*, 36(4), 924–935.

Luo, W., Huang, F., Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on Information Forensics and Security*, 5(2), 201–214.

Masnick, B., Wolf, J. (1967). On linear unequal error protection codes. *IEEE Transactions on Information Theory*, IT-3(4), 600–607.

Sarkar, A., Madhow, U., Manjunath, B.S. (2010). Matric embedding with pseudorandom coefficient selection and error correction for robust and secure steganography. *IEEE Transactions on Information Forensics and Security*, 5(2), 225-239.

Zhang, W., Zhang, X., Wang, S. (2007). A double layered "plus-minus one" data embedding scheme. *IEEE Signal Processing Letters*, 14(11), 848–851.

Zhang, X., Wang, S. (2006). Dynamical running coding in digital steganography. *IEEE Signal Processing Letters*, 13(3), 165–168.

Zhou, Z.D., Xu, C. (2005). An improved unequal error protection Turbo codes. In: *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 284–287.

Bahl, L.R., Cocke, J., Jelinek, F., Raviv, J. (1974). Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, IT-20, 284–287.

Q. Mao received the BS degree in Mechanical Engineering and Automation Science from Nanjing University of Aeronautics and Astronautics, Jiangsu, China, in 2000, and MS degree in Traffic Information Engineering and Control from Shanghai Ship and Shipping Research Institute, Shanghai, China, in 2003, and the PhD degree in Traffic Information Engineering and Control from Tongji University, Shanghai, China, in 2006. Since 2006, she has been with the faculty of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, where she is currently a lecturer. Her research interests include information security and coding.

C. Qin received the BS and ME degrees in electronic engineering from the Hefei University of Technology, Anhui, China, in 2002 and 2005 respectively, and the PhD degree in signal and information processing from Shanghai University, Shanghai, China, in 2008. Since 2008, he has been with the faculty of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, where he is currently a Lecturer. His research interests include image processing and multimedia security.

Nauja sparti klaidos nevienodos apsaugos schema skirta vaizdo stenografijai

Qian MAO, Chuan QIN

Straipsnyje nagrinėjama klaidos nevienodos apsaugos schema (UEP) skirta vaizdo stenografijai. Perduodant stenografijos vaizdus ryšio kanalais, paslėptas (embedded) pranešimas gali būti jautresnis triukšmams nei priedangos (cover) vaizdas. Pasiūlyta sparti užtriukšminto ryšio kanalo kodavimo schema, kuri užtikrina geresnę paslėpto pranešimo apsaugą ir blogesnę priedangos vaizdo apsaugą. Modeliavimo rezultatai rodo, kad ši kodavimo schema duoda skirtingus kodavimo proceso klaidos apsaugos lygius esant pastoviam kodavimo greičiui. Eksperimentiniai rezultatai parodė, kad paslėpti slaptieji vaizdai buvo geresnės kokybės nei priedangos vaizdai.