Towards Lower Bounds on Embedding Distortion in Information Hiding

Younhee Kim, Zoran Duric, and Dana Richards

George Mason University, Fairfax VA 22030, USA,
{ykim9, zduric, richards}@cs.gmu.edu

Abstract. We propose two efficient information hiding algorithms in the least significant bits of JPEG coefficients of images. Our algorithms embed information by modifying JPEG coefficients in such a way that the introduced distortion is minimized. We derive the expected value of the additional error due to distortion as a function of the message length and the probability distribution of the JPEG quantization errors of cover images. We have implemented our methods in Java and performed the extensive experiments with them. The experiments have shown that our theoretical predictions agree closely with the actual introduced distortions.

1 Introduction

1.1 Motivation

Surveillance video data are often used as evidence of traffic accident or crime. The surveillance system widely uses closed-circuit television (CCTV), which is exemplified by the small camera at ATM machines or parking lots. The CCTV system records scenes in analog film. Due to the high cost of film maintenance, the security industry seeks a way to replace it with digital system and store-in-files instead of films. However, there is one issue in using the digitally stored video as evidence: authentication. Because of the ease of undetectable alteration, it is essential to ensure that the video has not been tampered with after it was archived. Federal Rules of Evidence (FRE) states that the original of a recording is required to prove the content of the recording (FRE 1002) [1].

There is a requirement for authentication of digital image as evidence. An authentication system should detect any tampering in a marked image. It may be desirable for some applications to provide an indication of how much alteration has occurred and where the alteration has occurred. Another requirement is that the message extracting process should not require the original image. There are two technical approaches which provide authentication of digital video data: cryptographic approach [1, 2] and information hiding approach.

Cryptographic authentication, creates a digital digest of the original image and encrypts it with the signer's key, creating a digital signature. This digital signature [25] can be decrypted only by a key that is correspondent to the signer's key. The digital signature can prove data integrity: if the image is exactly the same as the original, the digest of the image will match the decrypted digest. In cryptographic authentication, the digital signature is attached with the original image or stored in a safe place for later use.

The information hiding approach inserts authentication data into the original image by modifying the image imperceptibly. Combining authentication data and image together is beneficial in many applications; however, the distortion caused by modifying the original image raises many concerns. The degradation in video quality is not noticeable by the human visual system; however, for example, it may affect image enhancement processing or a pattern-matching system in an attempt to recognize or identify a certain person appearing in the video. Such image processing algorithms require the highest possible image quality in order to work reliably. Law enforcement may ask questions such as "Is this image the same as what was originally captured?"

To overcome the concern of the information hiding approach to the authentication problem, we propose a embedding scheme to *minimize distortion due to embedding.* It maintains the high quality of image as well as combining the authentication data with to-be-authenticated data together, which makes the information hiding approach to authentication more attractive.

Distortion is a measure of the modification of the original data due to embedding information and it varies depending on the amount of information embedded in the image, which is called a *payload.* It is clear that a high payload increases the level of distortion. However, there has been very little work on how to optimally embed information in terms of the tradeoff between distortion and payload. We provide an analysis of distortion due to embedding with various payloads. This will allow users to achieve the maximum possible payloads with tolerable distortions of their data.

Most information hiding methods operate in two steps. First, a *cover object* is analyzed and the perceptually insignificant bits are identified. It is assumed that changing these bits will not make observable changes to the cover. Second, the identified bits are modified by the message bits to create a *stego object.* In this paper, cover object is an image in compressed JPEG [19] format. The perceptually insignificant bits correspond to a subset of LSBs of the JPEG coefficients. Although, the LSBs of JPEG coefficients are usually considered perceptually insignificant modifying some of these bits can produce detectable (but imperceptible) distortions of the original image. Our algorithms use parity codes and matrix-coding technique to minimize the distortion of the stego image relative to the cover image.

The paper is organized as follows. In Sec. 2 we briefly review the relevant prior work in the field. In Sec. 3 we provide technical background for our work including the basic facts about JPEG compression and the matrix coding. In Sec. 4 and 5 we describe our method and sketch the theoretical analysis of our method. In Sec. 6 we present some experimental results. Finally, in Sec. 7 we present the concluding remarks.

2 Related Work

With regard to the authentication that is based on information hiding, two problems are related: fragile watermarking and semi-fragile watermarking. In fragile watermarking, the inserted watermark is lost or altered as soon as any modification occurs in the cover object. Watermark loss or alteration indicates that the cover object has been tampered with, while the recovery of the watermark within the data indicates data originality. In semi-fragile watermarking, the inserted watermark is designed to be destroyed by some manipulations but to survive innocuous manipulations, e.g., moderate image compression. Since we are interested in authenticating the original data we will only discuss fragile watermarking.

The early fragile technique for authentication involves inserting the mark in the least significant bits (LSBs) of the actual image pixels [7, 8] and the added watermark is a pseudo-random sequence, which is not related to the content of the image. Wong [9] calculates a digest of the image using a hash function. The image ID, image size and user key are hashed and embedded by modifying the LSBs of pixels of the image. A hybrid method in color images was proposed by Yeung and Mintzer [10]; Fridrich and Goljan [11] proposed an improvement. Fragile watermarking systems in the transformation domain like JPEG have the advantage that the mark can be embedded in the compressed domain. Wu and Liu [29] describe a technique based on a modified JPEG encoder, which changes the quantized DCT coefficients before entropy coding. Kundur and Hatzinakos [21] and Xie and Arce [22] describe techniques based on the wavelet transform. Kundur modifies the Haar wavelet transformation coefficients while Xie modifies the SPIHT algorithm.

The goal of steganography is to insert a message into a carrier signal so that it cannot be detected by unintended recipients. Steganalysis attempts to discover hidden signals in suspected carriers or at the least detect which media contain hidden signals. Detailed survey of early algorithms and software for steganography and steganalysis can be found in [18, 28]. An early quantitative technique for steganalysis was designed by Westfeld and Pfitzmann [26]. This research prompted interest in both improving statistical detection techniques [13, 15] as well as building new steganographic methods that would be difficult to detect by statistical methods [27, 24, 16].

Various attempts have been made to make steganographic content difficult to detect, often by reducing the payload. Anderson and Petitcolas [3] suggested using the parity of a group of bits to encode a message bit; large groups of cover bits could be used to encode a single bit, the bits that need to be changed could be chosen in a way that would make detection hard. Westfeld [27] designed a steganographic algorithm $F5$ that uses matrix coding to minimize the modifications of the LSBs. Fridrich et al. [14, 15] reported several techniques for detecting steganographic content in images. If a message is inserted into the LSBs of an image, some features of the image change in a manner that depends on the message size.

Sallee [24] developed a hiding method that preserves distributions of individual JPEG coefficients. Fridrich et al. [16] have proposed an information hiding method that guarantees low distortion rates of stego objects. The method makes use of the JPEG quantization errors by computing all rounding errors of the JPEG coefficients. Note that for some coefficients the rounding error is $0.5 \pm \epsilon$. These coefficients can be rounded either down or up without a noticeable difference and they are considered changeable. Recently, Kim et al. [20] have described a parity-coding based hiding algorithm that minimizes distortion error by utilizing the rounding errors in JPEG quantization.

3 Technical Background

3.1 Information Hiding System

The goal of information hiding is to convey a message secretly and imperceptibly to people except a specific receiver. Generally, it modifies a cover object to embed message. We denote the cover object as a vector X and the message as M. M will be embedded in X by modifying X into $\hat{X}$, which is called a stego object.

$$X = (x_1, x_2, \ldots, x_l), \quad \hat{X} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_l), \quad M = (m_1, m_2, \ldots, m_k). \tag{1}$$

An information hiding algorithm has a pair of functions $\mathbf{f}$ and $\mathbf{g}$ such that

$$\hat{X} = \mathbf{f}(X, M), \quad M = \mathbf{g}(\hat{X}). \tag{2}$$

3.2 JPEG Image Format

We assume here that cover objects are image files in JPEG format, but our techniques are not limited to them. The JPEG image formatting removes some image details to obtain considerable saving of storage space without much loss of image quality. For the JPEG encoder, each channel is divided into 8×8 blocks and transformed using the two-dimensional discrete cosine transform (DCT). Let $f(i,j),\ i,j = 0, \ldots, N-1$ be an $N \times N$ image block in any of the channels and let $F(u,v),\ u,v = 0, \ldots, N-1$ be its DCT coefficient. See [17] for the mathematical specifics.

The coefficient $F(0,0)$ is the DC coefficient and all others are called AC coefficients. JPEG uses quantization and rounding formulas,

$$F'(u,v) = \frac{F(u,v)}{Q(u,v)}, \tag{3}$$

$$F''(u,v) = \mathit{Round}(F'(u,v)) \tag{4}$$

to obtain integer-valued coefficients $F''(u,v)$, where $Q(u,v)$ is a quantization table [17]. The process results in a quantization error:

$$\delta(u,v) = F''(u,v)Q(u,v) - F(u,v). \tag{5}$$

3.3 Minimizing Embedding Distortion

The cover object is obtained by a JPEG compression process and the JPEG coefficients and the corresponding rounding errors are known. Information hiding will add additional distortion beyond the quantization errors (see Eq. (5)).

Let X' and X'' be the vectors of DCT coefficients before and after the rounding, respectively (see Eq. (4)). The rounding error is given by $r_i = x''_i - x'_i$.

$$\begin{array}{l} X' = (x'_1, x'_2, \dots, x'_l). \\ X'' = \mathit{Round}(X') = (x''_1, x''_2, \dots, x''_l). \\ R = X'' - X' = (r_1, r_2, \dots, r_l). \end{array}$$

Our analysis will assume that each element of R is an independent, identically distributed (i.i.d.) random variable and that its probability density $p(r), r \in [-0.5, 0.5]$ is known. A message $M = (m_1 \;\; m_2 \dots m_k)$ is a binary sequence and each element, m_i, is a i.i.d. random variable. A message, M, is embedded into X, and the output of the embedding process is $\hat{X}$. In prior work, the cover object, X, was typically X'', but in this paper, X will be X'. Note that X' is only available during the JPEG encoding process.

We propose an embedding algorithm for minimizing distortion given rounding errors. We will show how bit-parity coding and matrix coding can be used to minimize the distortion.

4 Parity Coding

4.1 Embedding Algorithm

Our embedding algorithm makes use of given rounding errors. We seek a pair of functions $\mathbf{f}$ and $\mathbf{g}$ such that

$$\hat{X} = \mathbf{f}(X, R, M), \quad M = \mathbf{g}(\hat{X}) \tag{6}$$

and $\|\hat{X} - X\|_1$ is minimized. This approach assumes that encoding is done within JPEG, since R is known. Distortion is defined as:

$$D = |\hat{X} - X|. \tag{7}$$

Note that if $\hat{X} = X$ then $D = R$, i.e. if no information is embedded, the distortion is equivalent to the rounding error. Since embedding any message almost always requires changing bits, the best result that can be obtained is

$$\|D\|_1 \geq \|R\|_1.$$

We will show how bit-parity codes of length $n \leq l/k$ can be used to minimize the distortion $\|D\|_1$.

Embedding Algorithm The embedding process divides X into blocks of length n. To embed a bit m_i the block $X^i = x_{n(i-1)+1}, \ldots, x_{ni}$ is considered. If the parity of the LSBs of X^i is equal to m_i, no change needs to be made to any x_j, so $\hat{X}^i = X^i$. On the other hand, if the parity of the LSBs of X^i is different from m_i, we need to select an $x_j \in X^i$ to replace it by either $\hat{x}_j = x''_j - 1$ or by $\hat{x}_j = x''_j + 1$; in the first case, the distortion will be $d_j = -1 + r_j$ and in the second case, it will be $d_j = 1 + r_j$. One exception applies to this embedding algorithm. When $x''_j = \pm 1$, we will make $\hat{x}_j = \pm 2$ to avoid creating additional zero-valued coefficients. Since we are interested in minimizing $|d_j|$, we should use $\hat{x}_j$ that minimizes it, that is

$$\hat{x}_j = \begin{cases} 2, & r_j \geq 0 \ \ \& \ \ x''_j = 1 \\ x''_j - 1, & r_j \geq 0 \ \ \& \ \ x''_j \neq 1 \\ -2, & r_j < 0 \ \ \& \ \ x''_j = -1 \\ x''_j + 1, & r_j < 0 \ \ \& \ \ x''_j \neq -1. \end{cases}$$

In terms of rounding error, r_j, the distortion is given by

$$d_j = \begin{cases} 1 + |r_j|, & x''_j r_j > 0 \ \ \& \ \ x''_j = \pm 1 \\ 1 - |r_j|, & \text{otherwise.} \end{cases}$$

Finally, the additional error due to distortion, ε_j is given by

$$\varepsilon_j = \begin{cases} 1, & x''_j r_j > 0 \ \ \& \ \ x''_j = \pm 1 \\ 1 - 2|r_j|, & \text{otherwise.} \end{cases} \tag{8}$$

A goal in information hiding is to design embedding functions such that $\|d\|_1$ is minimal. Since r_js are already given, minimizing $\|d\|_1$ is equivalent to minimizing $\mathbf{\Delta} = \sum_{j=1}^{l} \varepsilon_j$.

4.2 Embedding Distortion

Let us define X_p as a set of the coefficients such that their corresponding embedding error is $\varepsilon_j = 1 - 2|r_j|$ and X_q as a set of the coefficients such that $\varepsilon_j = 1$.

$$\begin{aligned} X_p &= \{x_j | \ x''_j r_j \leq 0 \ \ \vee \ \ x''_j \neq \pm 1\} \\ X_q &= \{x_j | \ x''_j r_j > 0 \ \ \wedge \ \ x''_j = \pm 1\} \end{aligned}$$

Let $p = \frac{|X_p|}{|X_p| + |X_q|}$, i.e, the related proportion of all coefficients that belong to X_p and let $q = 1 - p$, i.e, the related proportion of X_q.

For a given block of coefficients, $X = \{x_1, \ \ldots, x_n\}$ of size n, there will be $0 \leq n_p \leq n$ coefficients from X_p and $n_q = n - n_p$ coefficients from X_q. For any n_p, the probability of the particular proportion of coefficients will be

$$P\{n_p = i\} = \binom{n}{i} p^i q^{n-i} \tag{9}$$

First, the distortion for those coefficients that belong to X_p is analyzed. We have assumed that r_js are i.i.d. random variables and that their probability density $f_r(x)$ is known. We can define this probability distribution for $\psi = |r|$ as

$$F_\psi(x) = \int_{-x}^{x} f_r(x)dx, \quad x \in [0, 0.5].$$

The probability distribution for $\nu = 1 - 2\psi$ is given by

$$F_\nu(x) = 1 - F_\psi(\frac{1-x}{2}), \quad x \in [0, 1]. \tag{10}$$

We are looking for a coefficient having the smallest embedding error within every block, X. If there are $n_p > 0$ coefficients from X_p in a given block, the algorithm will choose the coefficient corresponding to the minimal embedding error among the n_p coefficients. Since the embedding error for the coefficients from X_q is 1, which is always greater than the embedding errors of the coefficients from X_p, the remaining coefficients are not considered.

Given the probability distribution $F_\nu(x)$ for the embedding errors of the coefficients in X_p, the minimal additional error due to embedding is given by

$$\mu = \min_j\{\varepsilon_j\}, \;\; 1 \leq j \leq n.$$

The distribution of μ when $n_p = i$ is given by

$$F_\mu(x, i) = P_r\{\mu \leq x | n_p = i\} = \begin{cases} U(x-1), & i = 0 \\ 1 - (1 - F_\nu(x))^i, & i \geq 1, \end{cases} \tag{11}$$

where

$$U(x-1) = 0, \;\; x < 1$$
$$U(x-1) = 1, \;\; x \geq 1$$

and $F_\nu(x)$ is given by Eq. (10).

After taking account of all possible combinations of the coefficients, the distribution of additional error will be given by

$$F_\mu(x) = \textstyle\sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i} F_\mu(x, i). \tag{12}$$

The expected value of the embedding error will then be given by

$$E[\mu] = \textstyle\sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i} E[\mu | n_p = i], \tag{13}$$

where

$$E[\mu | n_p = i] = \int_0^\infty x dF_\mu(x, i), \;\; i \geq 0$$

5 Modified Matrix Coding

5.1 Background

Matrix coding was proposed by Crandall [12] to improve embedding efficiency by decreasing the number of required bit changes. Westfeld [27] proposed *F5*, a steganographic algorithm which implemented the matrix coding. In F5, cover data is the set of LSBs of quantized DCT coefficients after rounding. The notation $(1, n, k)$, where $n = 2^k - 1$, denotes embedding k message bits into an n bit sized block by changing only one bit of it. The embedding process divides X into blocks of length n and message data M into blocks of length k. To embed the i^{th} message block, $\{m_{k(i-1)+1}, \ldots, m_{ki}\}$, a cover data block $\{x_{n(i-1)+1}, \ldots, x_{ni}\}$ is used. Let us denote M and X as the message block and the cover block. The advantage of matrix coding is that we change only one bit to embed several bits. A function b needs to be defined in matrix coding:

$$b(X) = \bigoplus_{j=1}^{n} (x_j) \cdot j. \tag{14}$$

To calculate α, the position of the bit that needs to be changed, we use

$$\alpha = M \oplus b(X). \tag{15}$$

If $\alpha \neq 0$, then bit α in the block of X should be flipped, 1 to 0 or 0 to 1. The modified block is then given by

$$\hat{X} = \begin{cases} X, & \text{if } \alpha = 0. \\ x_1, \ldots, \neg x_\alpha, \ldots, x_n & \text{if } \alpha \neq 0. \end{cases} \tag{16}$$

On the decoder's side, k message bits are obtained from an n bit sized cover data by computing the following:

$$M = b(\hat{X}). \tag{17}$$

We cannot tune *F5* [27] to minimize distortion since the bit flip is completely constrained. We propose to modify $F5$ to increase the number of possible bit-change choices in each block. We describe our approach for two bit-changes. We call our method Modified Matrix Encoding (MME) and denote MME3, MME4 when we extend it to 3 and 4 bit-changes respectively.

5.2 Embedding Algorithm

MME will find pairs of numbers (β, γ) such that $\beta \oplus \gamma = \alpha$. If we use the embedding technique described in Sec. 4.1 for each cover block, X of length n, we are given two vectors of coefficients $(x'_1, ..., x'_n), (x''_1, ..., x''_n)$, before and after rounding respectively. We know the rounding errors $(r_1, ..., r_n)$ and the message block M of length k. As in Sec. 2 $X = X'$. We compute α using (15) and the pairs $(\beta_1, \gamma_1), ..., (\beta_h, \gamma_h)$ such that $\beta_i \oplus \gamma_i = \alpha$. Note that the number of pairs is $h = \frac{n-1}{2}$.

The embedding error using an unmodified $F5$ is given by $\varepsilon_0 = 1 - 2|r_\alpha|$, see (8). For each of the pairs (β_i, γ_i), the embedding error is given by one of four cases:

$$\varepsilon_i = \begin{cases} 2, & \text{if } x''_{\beta_i} r_{\beta_i} > 0 \ \& \ x''_{\gamma_i} r_{\gamma_i} > 0 \ \& \ x''_{\beta_i} = \pm 1 \ \& \ x''_{\gamma_i} \pm 1 \\ 2 - 2|r_{\gamma_i}|, & \text{if } x''_{\beta_i} r_{\beta_i} > 0 \ \ \& \ \ x''_{\beta_i} = \pm 1 \ \ \& \ \ x''_{\gamma_i} \neq \pm 1 \\ 2 - 2|r_{\beta_i}|, & \text{if } x''_{\gamma_i} r_{\gamma_i} > 0 \ \ \& \ \ x''_{\gamma_i} = \pm 1 \ \ \& \ \ x''_{\beta_i} \neq \pm 1 \\ 2 - 2(|r_{\beta_i}| + |r_{\gamma_i}|), & \text{otherwise.} \end{cases} \tag{18}$$

In order to decide how to create $\hat{X}$, we find

$$\mu = \min_j \{\varepsilon_j\}, \ \ 0 \leq j \leq \frac{n-1}{2}.$$

Given μ, we compute $\hat{X}$ by

$$\hat{X} = \begin{cases} X, & \text{if } \alpha = 0 \\ x_1, \ldots, \hat{x}_\alpha, \ldots, x_n, & \text{if } \mu = \varepsilon_0 \\ x_1, \ldots, \hat{x}_{\beta_i}, \ldots, \hat{x}_{\gamma_i}, \ldots, x_n, & \text{if } \mu = \varepsilon_i, \ i = 1, \ldots, \frac{n-1}{2}. \end{cases} \tag{19}$$

5.3 Embedding Distortion of MME

Let us denote X_p as a set of the coefficients such that their corresponding embedding errors will be $\varepsilon_0 = 1 - 2|r_j|$ when we change the coefficients. X_q is denoted as a set of the coefficients such that the embedding errors will be $\varepsilon = 1$ if we change the coefficients.

$$\begin{aligned} X_p &= \{x_j | \ x''_j r_j \leq 0 \ \ \vee \ \ x''_j \neq \pm 1\} \\ X_q &= \{x_j | \ x''_j r_j > 0 \ \ \wedge \ \ x''_j = \pm 1\} \end{aligned}$$

Let $p = \frac{|X_p|}{|X_p| + |X_q|}$, i.e, the related proportion of all coefficients that belong to X_p and let $q = 1 - p$, i.e, the related proportion of X_q. For a given block of coefficients, $X = x_1, \ \ldots, x_n$, of size n, the only cases we should care about are (a) $x_\alpha \in X_p \ \wedge \ (x_\beta \in X_p \ \wedge \ x_\gamma \in X_p)$ and (b) $x_\alpha \in X_q \ \wedge \ (x_\beta \in X_p \ \wedge \ x_\gamma \in X_p)$.

Let m be the number of pairs in which both x_β and x_γ are from X_p, $0 \leq m \leq h$. For any m, the probability of the particular proportion of coefficients will be

$$P\{m = i\} = \binom{h}{i} (p^2)^i (1 - p^2)^{h-i} \tag{20}$$

First, the distortion for coefficients that belong to X_p is analyzed. Again we assume that r_js are i.i.d. random variables and that their probability density $f_r(x)$ is known. Probability distribution for $y = |r|$ is given by

$$F_y(x) = \int_{-x}^{x} f_r(x) dx, \quad y \in [0, 0.5].$$

The probability density for $z = |r_1| + |r_2|$ is given by

$$f_z(x) = f_y(x) \bigotimes f_y(x), \quad z \in [0,1],$$

where $\bigotimes$ stands for convolution. The probability distribution is given by

$$F_z(x) = \int_0^z f_z(x)dx, \quad z \in [0,1].$$

The probability distribution for $\nu = 1 - 2y$ is given by

$$F_\nu(x) = 1 - F_y(\frac{1-x}{2}), \quad \nu \in [0,1]. \tag{21}$$

The probability distribution for $\omega = 2 - 2z$ is given by

$$F_\omega(x) = 1 - F_z(2-x), \quad \omega \in [0,2]. \tag{22}$$

The embedding error due to change of $x_\alpha \in X_p$ will follow the distribution of $F_\nu(x)$ and changes of x_β and x_γ will follow the distribution of $F_\omega(x)$.

To estimate the probability distribution of the embedding distortion due to embedding for (t,n,k) matrix codes, we use the order statistics [23]. As the first approximation, we have only considered the case when all embedding errors are $\varepsilon_i \leq 1$. Now, we need to obtain distribution of the smallest error we should take for embedding with consideration of embedding errors greater than 1.

The distribution of μ when $n_p = i$ and $x_\alpha \in X_p$ is given by

$$F_\mu(x,i) = P_{i,X_p}\{\mu \leq x | n_p = i\} = \begin{cases} U(x-1), & i = 0 \\ 1 - (1 - F_\nu(x))(1 - F_\omega(x))^i, & i \geq 1, \end{cases} \tag{23}$$

where $U(x)$ was defined in Sec. 4.

The distribution of μ when $n_p = i$ and $x_\alpha \in X_q$ is given by

$$F_\mu(x,i) = P_{i,X_q}\{\mu \leq x | n_p = i\} = \begin{cases} U(x-1), & i = 0 \\ 1 - (1 - F_\omega(x))^i, & i \geq 1. \end{cases} \tag{24}$$

After taking account of all possible combinations of the coefficients, the distribution of additional error will be given by

$$F_\mu(x) = \textstyle\sum_{i=0}^h \binom{h}{i} p^i q^{h-i} (p\ F_\mu(x,i,X_p) + q\ F_\mu(x,i,X_q)). \tag{25}$$

The expected value of embedding distortion due to embedding, $E[\mu]$, is given by

$$E[\mu] = \sum_{i=0}^h \binom{h}{i} p^i q^{h-i} p E[\mu|n_p = i, X_p]\ qE[\mu|n_p = i, X_q],$$

where

$$E[\mu|n_p = i, X_p] = \textstyle\int_0^\infty x dF_\mu(x,i,X_p),$$
$$E[\mu|n_p = i, X_q] = \textstyle\int_0^\infty x dF_\mu(x,i,X_q).$$

Since changes occur in $\frac{n}{n+1}$ cases in any block, the expected embedding error per bit is given by

$$E[\|\epsilon\|_1] = E[\mu] \times \frac{n}{n+1}.$$

6 Experimental Results

We have implemented our algorithms in Java. In this section we demonstrate the operation of our methods on two test images. Figure 1 shows the test images, which are color JPEG images. Rounding error histograms are also shown in Fig. 1; we estimate the rounding-error distributions by normalizing the histograms.

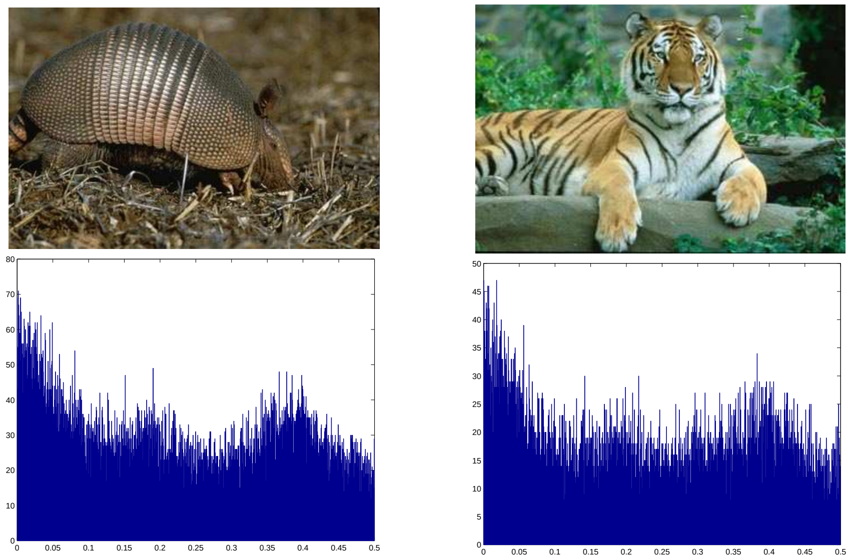

Fig. 1. Left column: armadillo image. Right column: tiger image. Top row: test images. Bottom row: rounding error histograms of the nonzero AC JPEG coefficients. The histogram is normalized to estimate a probability density of rounding errors.

The algorithm modifies a publicly available implementation of the JPEG image compression algorithm. After computing the DCT, all non-zero AC coefficients are marked for possible embedding and collected to form X'' with the corresponding rounding errors forming R. The implementation follows the algorithm described in Sec. 4 and 5. Our algorithm uses bit-parity and modified matrix encoding to choose the coefficients of which modifications introduce minimal embedding distortion.

All tests was accomplished with 7 different block-sizes for matrix coding $((t, 2^{k-1}, k)$, $k = 1, \ldots, 7)$ and for bit-parity coding $(2^k,\ k = 1, \ldots, 7)$.

Figures 2 and 3 show the theoretical embedding error analysis for parity coding and MME. They plot the comparison of the predicted embedding error to the real experimental embedding error and show close agreement between the theoretical prediction and the actual embedding distortion.

Figures 4 and 5 show the comparison of distortion in various embedding rates (μ^{-1}) using *F5*, *MME* and the extended versions of *MME*, MME3 and MME4,

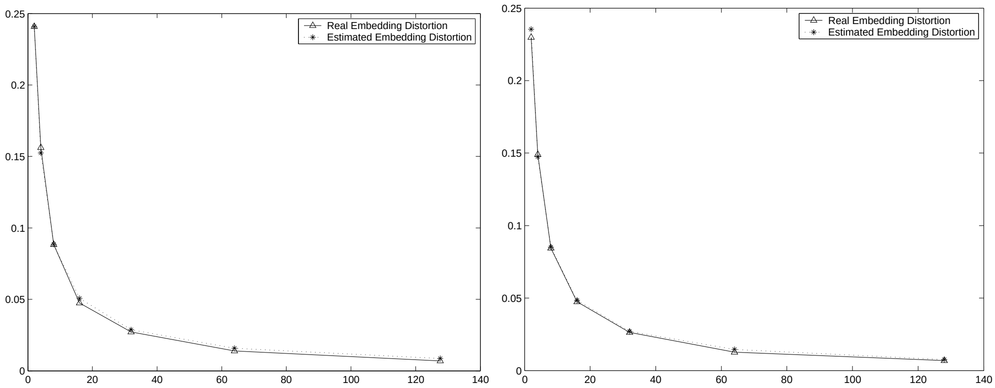

Fig. 2. Embedding error analysis of bit-parity coding in various block size n. Top: Theoretical embedding error and experimental embedding error for the armadillo image (left image in Fig. 1). Bottom: Comparison of the theoretical embedding error to the experimental embedding error for the tiger image (right image in Fig. 1).

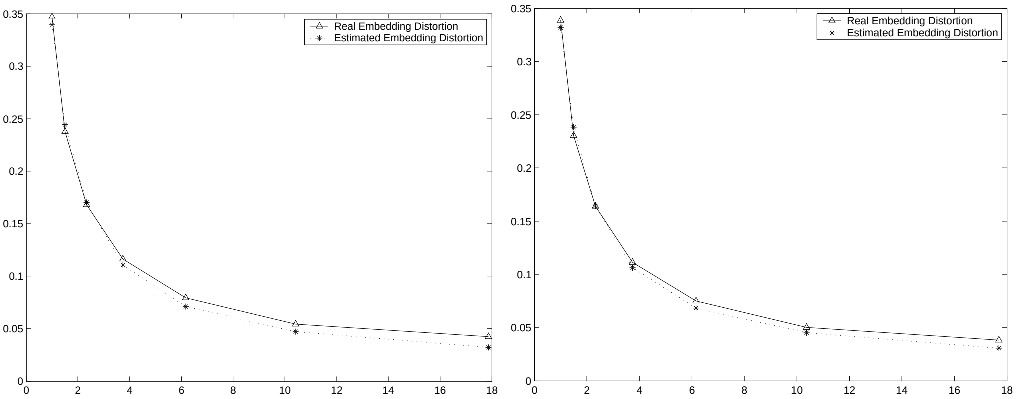

Fig. 3. Embedding error analysis of modified matrix encoding in embedding rates. X-axis is $\mu^{-1}1$. Top: result for for the armadillo image (left image in Fig. 1). Bottom: result for for the tiger image (right image in Fig. 1)

that modify up to 3 and 4 bits per block, respectively. (These algorithms are analyzed but not defined in this paper.) Note that the embedding errors caused by MME can be decreased by MME3 version noticeably, but benefit from MME4 is not much noticeable. The top graphs plot the distortions per embedding message bit in decreasing embedding rates, μ^{-1}. Note that the embedding rate is given by the block size divided by the number of message bits in the block. The bottom graphs plot the distortion per changed bit in decreasing embedding rates, μ^{-1}. The distortions due to our embedding algorithms are noticeably lower than one due to F5.

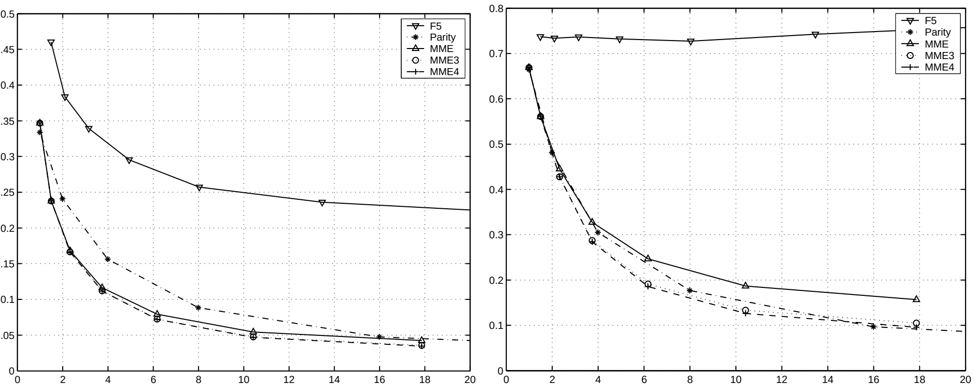

Fig. 4. Embedding error analysis for the armadillo image (left image in Fig. 1). Top row: Embedding distortion per embedding message bit with μ^{-1}. Bottom row: Embedding distortion per changing one bit with μ^{-1}.

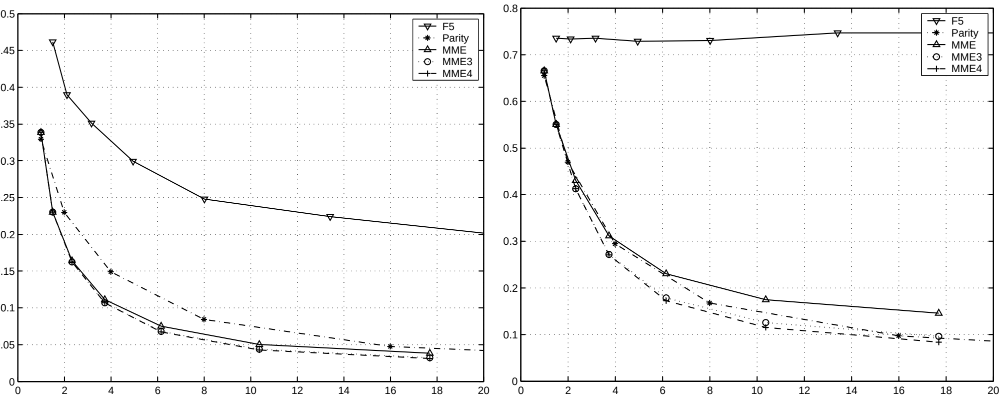

Fig. 5. Embedding error analysis for the tiger image (right image in Fig. 1). Top row: Embedding distortion per embedding message bit μ^{-1}. Bottom row: Embedding distortion per changing one bit μ^{-1}.

7 Conclusions

In this paper, we propose two efficient information hiding algorithms in the least significant bits of JPEG coefficients of images. Our algorithms embed information by modifying JPEG coefficients in such a way that the introduced distortion is minimized. We derive the expected value of the additional error due to distortion as a function of the message length and the probability distribution of the JPEG quantization errors of cover images. We have implemented our methods in Java and performed the extensive experiments with them. The experiments have shown that our theoretical predictions agree closely with the actual introduced distortions. Future work will include techniques for finding effective embedding algorithms using more sophisticated codes.

References

1. N. D. Beser, T. E. Duerr, and G. P. Staisiunas, "Authentication of digital video evidence," in *Proceedings of the SPIE International Conference on Applications of Digital Image Processing XXVI*, vol. 5203, November 2003, pp. 407–416.
2. A. Pramateftakis, T. Oelbaum, and K. Diepold, "Authentication of mpeg-4-based surveillance video," in *Proceedings of International Conference on Image Processing*, vol. 1, October 2004, pp. 33 – 37.
3. R. J.Anderson and F. A. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, May 1998.
4. P. Mouline and R. Koetter, "Data-hiding codes," *Proceedigns of the IEEE*, vol. 93, no. 12, pp. 2083–2126, December 2005.
5. F. Bartolini, A. Tefas, M. Barni, and I. Pitas, "Image authentication techniques for surveillance applications," *Proceedigns of the IEEE*, vol. 89, no. 10, pp. 1403–1418, December 2001.
6. E. T. Lin and E. J. Delp, "A review of fragile image watermarks," in *Proceedings of the Multimedia and Security Workshop Multimedia Contents*, October 1999, pp. 25–29.
7. T. A. Z. Van Schyndel, R G and C. F. Osborne, "A digital watermark," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, November 1994, pp. 86–90.
8. R. Wolfgang and E. Delp, "A watermark for digital images," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 3, September 1996, pp. 219–222.
9. P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for imageauthentication and ownership verification," *IEEE Transactions on Image Processing*, vol. 10, no. 10, 2001.
10. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, October 1997, pp. 680 – 683.
11. J. Fridrich and M. G. andArnold C.Baldoza, "New fragile authentication watermark for images," in *Proceedings of IEEE Internatinal Conference on Image Processing*, September 2000, pp. 446–449.
12. R. Crandall. "Some Notes on Steganography." Posted on Steganography Mailing List, 1998. http://os.inf.tu-dresden.de/ westfeld/crandall.pdf

13. J. Fridrich. ”Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes.” *Proc. 6th Information Hiding Workshop*, Toronto, Canada, 2004.
14. J. Fridrich, M. Goljan, and R. Du. “Detecting LSB Steganography in color and gray-scale images”, *IEEE Multimedia Magazine,* pp. 22–28, October 2001.
15. J. Fridrich, M. Goljan, and D. Hogea. “Steganalysis of JPEG images: Breaking the F5 algorithm”, LNCS 2578, Springer-Verlag, Berlin Heidelberg, pp. 310–323, 2002.
16. J. Fridrich, M. Goljan, and D. Soukal. “Perturbed quantization steganography with wet paper codes.” *Proc. ACM Multimedia Workshop*, Magdeburg, Germany, 2004.
17. R.C. Gonzales, R.E. Woods, “Digital Image Processing”, Addison-Wesley, 2002
18. N. Johnson, Z. Duric, and S. Jajodia. *Information Hiding: Steganography and Watermarking — Attacks and Countermeasures.*, Kluwer Academic Publishers, Boston, 2000.
19. Joint Photographic Experts Group. http://www.jpeg.org/public/jpeghomepage.htm.
20. Y. Kim, Z. Duric, D.Richards. “Limited Distortion in LSB Steganography.” *Proc. SPIE Electronic Image*, 2006
21. D. Kundur and D. Hatzinakos, “Towards a telltale watermarking technique for tamper-proofing,” in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, October 1998, pp. 409–413.
22. G. Liehua and X. Arce, “Joint wavelet compression and authentication watermarking,” in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, October 1998, pp. 427–431.
23. A. Paopoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, Boston, MA, 1991.
24. P. Sallee. ”Model-based steganography.” *Proc. Information Hiding Workshop*, LNCS 2939, Springer-Verlag, Berlin, pp. 154–167, 2003.
25. B. Schneier, *Applied Cryptography.* John Willey & Sons. Inc., 1996.
26. A. Westfeld and A. Pfitzmann. “Attacks on steganographic systems.” *Proc. Information Hiding Workshop*, LNCS 1768, Springer-Verlag, New York, pp. 61-75, 1999.
27. A. Westfeld. “F5—a steganographic algorithm: High capacity despite better steganalysis.” *Proc. Information Hiding Workshop*, LNCS 2137, Springer-Verlag, Berlin, pp. 289-302, 2001.
28. P. Wayner. *Disappearing Cryptography.* 2nd ed., Morgan Kaufmann, San Francisco, 2002.
29. M. Wu and B. Liu, “Watermarking for image authentication,” in *Proceedings of IEEE Internatinal Conference on Image Processing*, vol. 2, October 1998, pp. 437–441.