Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Data embedding in digital images using critical functions

Xin Liao [a,b,*], Zheng Qin [a], Liping Ding [c]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[b] Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA
[c] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Keywords:
Image steganography
Uniform embedding
Adaptive embedding
Steganographic embedding function
Pixel correlation function

ABSTRACT

In this paper, "uniform embedding" (independent of image contents and pixel correlations while embedding) and "adaptive embedding" (depend on image contents and pixel correlations while embedding) in image steganography are investigated. A compact steganographic embedding function is proposed to ensure the correctness and efficiency, and a pixel correlation function is utilized to discriminate the image smoothness. Two feasible image steganographic frameworks using these critical functions are presented, and some well-known image steganographic methods can be derived from the proposed frameworks. The effectiveness of the proposed frameworks is experimentally validated by constructing and testing some special data hiding methods in the case of four neighboring pixel as a processing unit. Experimental results show that the proposed methods can achieve better visual performance and statistical undetectability compared with the prior works. Another promising merit of our work is the potential to provide steganographers general-purpose strategies to acquire new image steganographic methods.

1. Introduction

Steganography is a technique of covert communication [1]. It aims to embed secret messages into an innocent carrier signal by slightly altering its most insignificant components, such that an unauthorized user will not be aware of the existence of secret data [2]. A good steganographic method should have good visual/statistical imperceptibility and a sufficient payload. The former is essential for the security of covert communication and the latter ensures that a large quantity of secret data can be conveyed [3].

Lots of practical image steganographic embedding methods apply a mutually independent embedding operation to all or selected elements of the cover image. The most common and well-known method falling under this paradigm is called least significant bit (LSB) substitution, embedding secret data by replacing fixed-length LSBs of a cover pixel with secret bits directly [4]. Chan et al. proposed a simple and efficient optimal pixel adjustment process (OPAP) method to improve LSB substitution. The basic concept of it is to increase or decrease the most significant bit part by 1 in order to reduce the embedding distortion [5]. LSB matching (LSBM) employs a minor modification to LSB substitution. If the secret bit does not match the LSB of the cover pixel, then one is randomly either added or subtracted from the value of the cover pixel [6]. Unlike the above mentioned methods, some steganographic methods employ two pixel pair as an embedding unit. Mielikainen presented the LSB matching revisited method (LSBMR), which allows embedding the same payload as LSBM but with fewer changes to the cover image [7]. Chao et al. proposed a novel data hiding method based on the diamond encoding (DE), and one secret k-ary digit is concealed into the diamond characteristic value of two pixel pair [8]. In 2012, Hong et al. proposed a novel data embedding method called APPM by providing a specially designed neighborhood set, which is employed to embed message digits with a smallest notational system [9]. Besides that, some other steganographic techniques using a block of more than two pixels as an embedding unit have been proposed. Exploiting modification direction (EMD) proposed by Zhang and Wang is an efficient steganographic technique, and $\log_2(2n+1)$ secret bits are embedded into n cover pixels and at most only one pixel is increased or decreased by 1 [10]. To highlight the pixel modification directions and achieve high embedding capacity, Sun et al. presented an improved method named HoEMD, in which a pixel with a larger change implies more pixel directions [11].

Note that all the above steganographic methods are independent of image contents and pixel correlations. When applying steganographic methods to the cover image, an equal amount of secret bits is concealed

[*] Corresponding author at: College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China.
E-mail address: xinliao@hnu.edu.cn (X. Liao).

http://dx.doi.org/10.1016/j.image.2017.07.006
Received 2 December 2016; Received in revised form 20 July 2017; Accepted 25 July 2017
Available online 31 July 2017
0923-5965/© 2017 Elsevier B.V. All rights reserved.

into each pixel and an equal degree of embedding distortion is caused. Therefore, these steganographic methods can be called "Uniform Embedding". In fact, digital images exhibit quite complex statistical dependencies among individual pixels, and not all pixels in a cover image can tolerate equal amount of changes without causing noticeable distortion. According to the characteristics of Human Visual System (HVS), it is sensitive to the changes in the pixels of the smooth areas, while it is not sensitive to changes in the edge areas. Therefore, some steganographic methods called "Adaptive Embedding" have been proposed in which the amount of bits to be embedded in each pixel is variable.

"Adaptive Embedding" methods can also be divided into several categories according to the number of pixels in each embedding unit. In the first category, these steganographic methods embed variable bits into each pixel. Chang and Tseng utilized the side information, i.e., the difference between the pixel and its upper and left side pixels, to decide the number of bits to be embedded [12]. Zhang and Wang proposed a multiple base notational system (MBNS) steganographic method, in which secret data are first transformed into symbols in a notational system with multiple bases and the amount of information carried by each pixel is adapted to the pixel value variation in the immediate neighborhood [13]. In the second category, these steganographic methods process a two-pixel pair at a time. Wu and Tsai presented an adaptive steganographic method using pixel-value differencing (PVD), where the number of bits to be embedded in a pixel pair is decided by the difference value between two neighboring pixels [14]. Wu et al. combined it and LSB later [15]. Wang et al. presented a steganographic method with the modulus function (MF-PVD), utilizing the remainders of two consecutive pixels to record the information of secret data [16]. Yang et al. proposed an edge adaptive steganographic method (EA-PVD), using the difference value of two consecutive pixels to distinguish between edge areas and smooth areas. All pixels are embedded by the k-bit OPAP method, where k is decided by the level which the difference value belongs to [17]. In 2011, Luo et al. expanded LSBMR method and selected the embedding regions according to the size of secret messages and the differences between two consecutive pixels (LSBMR-PVD) [18]. Sun et al. proposed an adaptive EMD method with the consideration of HVS (AdEMD) [11]. Hong et al. embedded data into two-pixel pairs using DE method and concealed digits in multiple-base according to the corresponding pixel-value differencing (DE-PVD) [19]. Shen et al. utilized an optimization problem to minimize the embedding distortion of the pixel [20]. Recently Hussaina et al. combined parity-bit pixel value differencing with improved rightmost digit replacement [21]. Note that the features of image edge can be considered sufficiently by using multi-pixel blocks [22]. Liu et al. proposed two generalizations of pixel-value differencing for data hiding (G-PVD). In each n-pixel block, $n-1$ differences are calculated between consecutive pixels, and then more differences can be used to hide secret data [23]. Yang et al. proposed a steganographic method using four-pixel differencing and pixel-value shifting operations (FPVD) [24], and it was improved by using modulus function and optimization theory later (MF-FPVD) [25]. In 2011, the authors introduced the average differencing of four-pixel values to design efficient steganographic methods, and then proposed two novel methods based on OPAP and EMD, respectively (OPAP-ADFPV and EMD-ADFPV) [26,27]. An octonary-PVD method with 3×3 pixel block was designed, in which the number of bits to be embedded in each pixel was decided by its neighbors in eight directions [28]. Chen et al. proposed a novel image steganographic method with 2×2 pixel block, and secret data was randomly embedded instead of sequential hiding [29]. In fact, designing adaptive steganographic schemes can be formulated as a minimal distortion framework for the entire image pixels. A distortion function is firstly built to decide the probable embedding change positions adaptively, and then combined with the advanced syndrome-trellis coding technique [30]. The distortion function of HUGO (highly undetectable stego) [31] computed the weighted sum of differences between the feature vectors respectively extracted from cover and stego images. WOW (wavelet obtained weights) [32], S-UNIWARD (spatial-universal wavelet relative distortion) [33], HILL (high-pass, low-pass, and low-pass) [34] designed the distortion function based on diverse image filters. MG (multivariate Gaussian) [35] was the first model driven framework to obtain the distortion, which was subsequently extended by utilizing a better variance estimator and the multivariate generalized Gaussian model (MVGG) [36]. CMD (clustering modification directions) [37] and Synchronize [38] strategies could preserve the correlation between neighboring pixels, which could be applied together with the above-mentioned distortion functions.

In this paper, two practical and efficient image steganography frameworks for "uniform embedding" and "adaptive embedding" are investigated, respectively. We elaborate the steganographic embedding function and the pixel correlation function in a more systematic manner, guaranteeing the correctness and effectiveness of the proposed frameworks. Our contributions can be summarized in the following aspects.

(1) Two novel functions (the steganographic embedding function and the pixel correlation function) and their characteristics are investigated, and the new "uniform embedding" and "adaptive embedding" frameworks are proposed based on these critical functions.

(2) The effectiveness of two proposed frameworks is experimentally validated by constructing the special image steganographic methods in the case of four neighboring pixel as a processing unit, and showing improvements in visual performance and statistical undetectability.

(3) The proposed frameworks are flexible and general-purpose. Some well-known image steganographic methods can be obtained from the proposed frameworks. Once the embedder specifies the constructions of the critical functions by himself, the proposed frameworks provide essential tools for constructing practical image steganographic methods.

The remainder of this paper is organized as follows. In Section 2, a steganographic embedding function F along with its characteristics are introduced, and then a "uniform embedding" framework is proposed. The pixel correlation function G is described and analyzed in Section 3, and a novel "adaptive embedding" framework is designed based on two critical functions. Section 4 presents investigative experiments and analysis aimed at comparing the performances among some state-of-the-art methods and the proposed frameworks in the case of four neighboring pixel as a processing unit. We give further discussions about how to construct a new image steganographic method using the proposed general-purpose strategies in Section 5. Finally, the conclusions are made.

In this paper, vectors will be typeset in boldface and their individual elements with the corresponding lower-case letters in italics. We utilize the symbols $\mathbb{Z}$ and $\mathbb{Z}^+$ to represent the set of all integers and positive integers, and $\mathbb{N}$ is used to represent the set of positive integers and zero. For any vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}^n$, $\|\mathbf{x}\|_l$ is applied to represent the l-norm, $\|\mathbf{x}\|_l = (\sum_{i=1}^{n} |x_i|^l)^{\frac{1}{l}}$. For $l = 1$ we get the taxicab norm, for $l = 2$ we get the Euclidean norm, and as l approaches ∞ the l-norm approaches the Maximum norm, i.e., $\|\mathbf{x}\|_\infty = \max(|x_1|, |x_2|, \ldots, |x_n|)$.

2. "Uniform Embedding" Image Steganography Framework

In this section, a steganographic embedding function F is derived for efficient steganographic embedding, and a "uniform embedding" framework is proposed based on it. Good function characteristics can ensure the correctness and effectiveness of the proposed framework. Some previous steganography methods can be regarded as special and variational cases, i.e., they are equivalent to the proposed framework if and only if the construction of F is modified.

2.1. The steganographic embedding function F

In this subsection, a steganographic embedding function F is proposed for efficient steganographic embedding. Inspired by refs. [8,9,11,12,16], the critical function F is designed under the modular arithmetic. The modulus function will be applied to adjust the remainder of pixel values for data embedding. For the modulus number b, it could accommodate b difference varieties, and then $\lfloor \log_2 b \rfloor$ secret bits can be embedded at a time. Furthermore, by using the modulus operation, the adjustment range of the weight summation is reduced to $[-b/2, b/2]$, and then the pixel modifications would be potentially decreased.

Let $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{Z}^n$, $\mathbf{A} = (a_1, a_2, \dots, a_n) \in \mathbb{N}^n$ and $k \in \mathbb{N}$. The steganographic embedding function F is derived as follows.

$$F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b,$$
$$\|\mathbf{x}\|_1 \le k,\ a_i = (2k+1)^{i-1},\ b = 1 + \sum_{i=1}^{n} \binom{n}{i}\binom{k}{i} \times 2^i. \tag{1}$$

Characteristic 1: The number of mutually exclusive $\mathbf{x}$ satisfying $\|\mathbf{x}\|_1 \le k$ is b altogether.

Proof. We firstly assume that $\|\mathbf{x}\|_1 = l$ $(1 \le l \le k, l \in \mathbb{Z}^+)$, and the number of zero values of x_i in $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is m $(0 \le m \le n-1, m \in \mathbb{N})$. Thus, we have $\binom{n}{m}\binom{l-1}{n-m-1}\times 2^{n-m}$ mutually exclusive vectors $\mathbf{x}$ according to permutations and combinations.

Along with changing m, there are altogether $\sum_{m=0}^{n-1}\binom{n}{m}\binom{l-1}{n-m-1} \times 2^{n-m}$ mutually exclusive vectors $\mathbf{x}$ such $\|\mathbf{x}\|_1 = l$ holds.

The total number of mutually exclusive vectors consist of the summation of the above equation from $l = 1$ to k and the special case that all x_i are zeros. Thus, we have

$$\begin{aligned}
&1 + \sum_{l=1}^{k}\sum_{m=0}^{n-1}\binom{n}{m}\binom{l-1}{n-m-1} \times 2^{n-m} \\
&= 1 + \sum_{m=0}^{n-1}\binom{n}{m}\sum_{l=1}^{k}\binom{l-1}{n-m-1} \times 2^{n-m} \\
&\overset{\textcircled{1}}{=} 1 + \sum_{m=0}^{n-1}\binom{n}{m}\binom{k}{n-m} \times 2^{n-m} \\
&\overset{\textcircled{2}}{=} 1 + \sum_{m=0}^{n-1}\binom{n}{n-m}\binom{k}{n-m} \times 2^{n-m} \\
&= 1 + \sum_{i=1}^{n}\binom{n}{i}\binom{k}{i} \times 2^{i} \quad (n-m \triangleq i) \\
&= b.
\end{aligned}$$

Here ① and ② are by the combinatorial identity $\sum_{l=1}^{k}\binom{l-1}{n-m-1} = \binom{k}{n-m}$ and $\binom{n}{m} = \binom{n}{n-m}$, respectively.

Characteristic 2: It will be different function values when the input vectors are mutually exclusive, i.e., $F(\mathbf{x}) \ne F(\mathbf{x}')$ if $\mathbf{x} \ne \mathbf{x}'$.

Proof. We firstly assume two mutually exclusive vectors $\mathbf{x} = (x_1, x_2, \dots, x_n)$ and $\mathbf{x}' = (x'_1, x'_2, \dots, x'_n)$ have the same function values $F(\mathbf{x}) = F(\mathbf{x}')$, and j is the largest index such that $x_j \ne x'_j$. Without loss of generality, suppose $x_j > x'_j$. Calculate the difference using modular arithmetic

$$\begin{aligned}
(F(\mathbf{x}) - F(\mathbf{x}')) \bmod b &= \left(\sum_{i=1}^{j} a_i x_i \bmod b - \sum_{i=1}^{j} a_i x'_i \bmod b\right) \bmod b \\
&= \sum_{i=1}^{j} a_i (x_i - x'_i) \bmod b.
\end{aligned}$$

Note that $(2k+1)^{j-1} = 2k\sum_{i=1}^{j-1}(2k+1)^{i-1} + 1$, we have $a_j = 2k\sum_{i=1}^{j-1} a_i + 1$. Therefore,

$$\begin{aligned}
a_j(x_j - x'_j) &\ge a_j \\
&= 2k\sum_{i=1}^{j-1} a_i + 1 \\
&\overset{\textcircled{3}}{\ge} (x'_i - x_i)\sum_{i=1}^{j-1} a_i + 1 \\
&> \sum_{i=1}^{j-1} a_i(x'_i - x_i).
\end{aligned}$$

Here ③ is by the constraints $\|\mathbf{x}\|_1 \le k$ and $\|\mathbf{x}'\|_1 \le k$. From the above, we know that $a_j(x_j - x'_j) - \sum_{i=1}^{j-1} a_i(x'_i - x_i) > 0$, i.e., $\sum_{i=1}^{j} a_i(x_i - x'_i) > 0$.

Note that both $F(\mathbf{x})$ and $F(\mathbf{x}')$ belong to $[0, b-1]$, we have $F(\mathbf{x}) \ne F(\mathbf{x}')$. This contradicts the assumption. Therefore, two mutually exclusive vectors have different function values.

2.2. The proposed "uniform embedding" framework

In this subsection, a "Uniform embedding" framework is proposed based on the steganographic embedding function F. The proposed "uniform embedding" framework employs n pixels as an embedding unit to conceal the secret data. The detailed embedding process and extraction process are given as follows.

2.2.1. Embedding process

All the pixels in the cover image are gray values with the range of $[0, 255]$. The cover image is partitioned into non-overlapping n-pixel blocks, and their corresponding gray values are $\mathbf{p} = (p_1, p_2, \dots, p_n)$. The parameters in Eq. (1) are predefined by users. For each n-pixel block, $\lfloor \log_2 b \rfloor$ secret bits will be embedded, and the detailed steps are as below.

Step 1: Calculate the function value $F(\mathbf{p}) = \mathbf{A}\mathbf{p}^T \bmod b$.

Step 2: Read $\lfloor \log_2 b \rfloor$ bits from the binary secret bit stream, and then transform into decimal value s. Calculate $d = s - F(\mathbf{p}) \bmod b$. If $d = 0$, no modification for the n pixels is needed. Secret data have been embedded, and then run the next n-pixel block. Otherwise, go to the next step.

Step 3: Search a vector $\mathbf{x}$ such that $F(\mathbf{x}) = d$ and $\|\mathbf{x}\|_1 \le k$. Since the function F is one-to-one function, we can find the unique vector $\mathbf{x}$. According to the detailed analysis in Section 2.4, the computation complexity is acceptable even if the exhaustive searching is used.

Step 4: Calculate $\mathbf{p}' = (p'_1, p'_2, \dots, p'_n)$ by $\mathbf{p}' = \mathbf{p} + \mathbf{x}$. Adjust $\mathbf{p}'$ to $\mathbf{p}'' = (p''_1, p''_2, \dots, p''_n)$ when one stego-pixel value has the overflow or underflow problem, i.e., the resulting pixel value falls outside the gray value range $[0, 255]$. If $p'_i > 255$, then $p''_i = p'_i - b$. If $p'_i < 0$, then $p''_i = p'_i + b$. If $0 \le p'_i \le 255$, then $p''_i = p'_i$.

Step 5: Replace $\mathbf{p}$ by $\mathbf{p}''$ and modify the pixel values. The purpose of $\lfloor \log_2 b \rfloor$-bit secret data hiding have been achieved.

2.2.2. Extraction process

In the extraction process, we can quickly extract secret data without the original image. Partition the stego image into n-pixel blocks, which is identical with the embedding procedure. For each n-pixel block, their corresponding gray values are $\mathbf{p}'' = (p''_1, p''_2, \dots, p''_n)$. Calculate the function value $F(\mathbf{p}''')$, and transform into the binary secret bit stream with the length of $\lfloor \log_2 b \rfloor$.

2.3. A simple example

We will use a simple example to illustrate the proposed framework clearly, as shown in Fig. 1. Suppose the parameters of the proposed framework are $k = 2$ and $n = 4$. The corresponding gray values of four-pixel block are (170,171,174,170). The secret bits to be embedded are 10101_2, i.e., the decimal value $s = 21$. According to Eq. (1), we have $b = 41$. Calculate the function value $F(170, 171, 174, 170) = 16$, and then $d = (s - 16) \bmod 41 = 5$. The suitable vector $\mathbf{x} = (0, 1, 0, 0)$ satisfies $F(\mathbf{x}) = 5$. Thus, replace $(170, 171, 174, 170)$ by $(170, 172, 174, 170)$. The secret data can be extracted without the original image. Extract the embedded digit $s = F(170, 172, 174, 170) = 21$, so we obtain the embedded secret bits 10101_2.

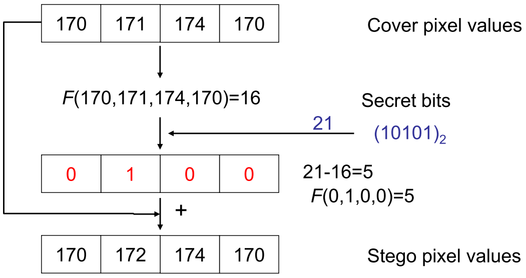

Fig. 1. A simple example of the proposed "Uniform Embedding" framework.

2.4. Analysis

Since the function F is a one-to-one function, the correctness and feasibility of the proposed framework can be explicitly validated. $F(\mathbf{p}'') = \mathbf{A}\mathbf{p}''^T \bmod b = (\mathbf{A}\mathbf{p}^T + \mathbf{A}\mathbf{x}^T) \bmod b = (F(\mathbf{p}) + F(\mathbf{x})) \bmod b = (F(\mathbf{p}) + d) \bmod b = s$, the receiver can extract the secret data exactly.

According to the embedding procedure, the modification of pixel values is decided by the vector $\mathbf{A} = (a_1, a_2, \dots, a_n)$ and the modulus b. The maximal accumulative modification for n-pixel block $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is k because of $\|\mathbf{x}\|_1 \le k$. Furthermore, the amount of secret bits embedded into the n-pixel block is $\lfloor \log_2 b \rfloor$. Thus, for a given n, the ratio between the embedding capacity and the embedding distortion is

$$R(n) = \frac{\lfloor \log_2 b \rfloor}{k} = \frac{\lfloor \log_2 (1 + \sum_{i=1}^{n} \binom{n}{i}\binom{k}{i} \times 2^i) \rfloor}{k}. \tag{2}$$

Specially, $R(2) = \lfloor \log_2(2k^2 + 2k + 1) \rfloor / k$, $R(3) = \lfloor \log_2(4k^3 + 6k^2 + 8k + 6)/3 \rfloor / k$ and $R(4) = \lfloor \log_2(2k^4 + 4k^3 + 10k^2 + 8k + 3)/3 \rfloor / k$. We can obtain larger embedding capacity by increasing the parameter k, but the image quality and attack-resistance would be decreased. In practical application, we should determine the suitable parameters based on numerous experiment simulations, guaranteeing the proposed framework can provide enough capacity while maintaining the smallest image distortions and highest statistical undetectability.

In the proposed "Uniform Embedding" framework, secret bits will be embedded by skillfully modifying the pixel values. The steganographer is require to do some basic arithmetic operations of n-dimensional vector, such as $F(\mathbf{p}) = \mathbf{A}\mathbf{p}^T \bmod b$, $d = s - F(\mathbf{p}) \bmod b$, and $\mathbf{p}' = \mathbf{p} + \mathbf{x}$. The computation complexity of these operations is no more than $O(n)$. The maximal accumulative modification is k. For a given $k_0 \le k$, we try to find a vector $\mathbf{x}$ such that $\mathbf{A}\mathbf{x}^T \bmod b = d$ and $\|\mathbf{x}\|_1 = k_0$. The maximum searching range of each x_i is $2k_0$, and the computational cost is no more than $O(2k_0 * n)$ even if the exhaustive searching is used. Thus, the computation cost of the embedding process is $O(2kn + 2(k-1)n + 2(k-2)n + \cdots + 2n) = O(nk(k+1)) = O(nk^2)$. The receiver would calculate $F(\mathbf{p}'') = \mathbf{A}\mathbf{p}''^T \bmod b$, and transform it into the binary secret bit stream, so the computation complexity is $O(n)$. Thus, the computation complexity of the proposed framework is $O(nk^2)$, which would be mainly determined by the parameter k. When k is small, the proposed framework can be solved in polynomial time. For example, for the proposed framework in the case of $n = 4$ and $k = 2$, we conduct the experiments on desktop PC running Windows 7 Professional with 16GB memory and 3.6 GHz Intel(R) Core(TM) i7-4790 processor. The time cost for each image is less than 0.5 s. Specifically, for the classic image "Lena", it will cost only 0.25 s.

2.5. Discussions

After further study, it is shown that some previous steganographic methods can be regarded as the special and equivalent versions of the proposed framework. That is to say, the following image steganographic methods can be obtained only by modifying the construction of the function F.

DE [8]: DE method executes two-pixel pair at a time, and is equivalent to the proposed framework in the case of $n = 2$. The coefficients of the function F is changed to $a_1 = 1$, $a_2 = 2k + 1$ and $b = 1 + 2k + 2k^2$.

$$\begin{aligned} &F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b, \\ &\|\mathbf{x}\|_1 \le k,\ a_i = (2k+1)^{i-1},\ b = 1 + 2k + 2k^2. \end{aligned} \tag{3}$$

APPM [9]: APPM method executes two-pixel pair at a time, and the coefficients of the function F are required to be computed firstly. For $a_1 = 1$ and the given integer b, we calculate a_2 by solving an optimization problem. APPM restricts the maximal modification of each pixel $\|\mathbf{x}\|_\infty$ to be as small as possible.

$$\begin{aligned} &F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b, \\ &\|\mathbf{x}\|_\infty \text{ as small as possible},\ a_1 = 1. \end{aligned} \tag{4}$$

EMD [10]: EMD method executes n-pixel block at a time, and the coefficients of the function F is changed to $a_i = i$ and $b = 2n + 1$. The total modification of each block is not more than 1 ($\|\mathbf{x}\|_1 \le 1$).

$$\begin{aligned} &F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b, \\ &\|\mathbf{x}\|_1 \le 1,\ a_i = i,\ b = 2n + 1. \end{aligned} \tag{5}$$

HoEMD [11]: HoEMD method executes n-pixel block at a time, and the coefficients of the function F are replaced by $a_i = (2k+1)^{i-1}$ and $b = (2k+1)^n$. HoEMD limits the maximal modification of each pixel, i.e., the constraint is changed to $\|\mathbf{x}\|_\infty \le k$.

$$\begin{aligned} &F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b, \\ &\|\mathbf{x}\|_\infty \le k,\ a_i = (2k+1)^{i-1},\ b = (2k+1)^n. \end{aligned} \tag{6}$$

3. "Adaptive Embedding" Image Steganography Framework

In this section, a critical function G (named "pixel correlation function") is derived for discriminating the image smoothness, and an "adaptive embedding" framework is proposed based on the functions F and G.

3.1. The pixel correlation function G

The pixel correlation function G is full exploited to classify a n-pixel block $(x_1, x_2, \dots, x_n)$ as a smooth area or an edge area.

$$G(x_1, x_2, \dots, x_n) : [0, 255]^n \to [0, \infty). \tag{7}$$

According to the predefined thresholds, the function value $G(x_1, x_2, \dots, x_n)$ is divided into different levels. It is obvious that the image smoothness can be estimated more exactly as n increases.

Most of existing "adaptive embedding" methods employ two neighboring pixels as an embedding unit, and use the difference value of two consecutive pixels to distinguish between edge areas and smooth areas [11,14–19,21]. It can be regarded as the function G in the case of $n = 2$.

$$G(x_1, x_2) = |x_1 - x_2|. \tag{8}$$

where x_1 and x_2 are two neighboring pixels values.

The difference value of neighboring pixels could be generalized by considering n-pixel block.

$$\begin{aligned} &G(x_1, x_2, \dots, x_n) = \sum_{x_i \in S} (x_i - x_{\min})/(n-1), \\ &x_{\min} = \min\{x_1, x_2, \dots, x_n\},\ S = \{x_1, x_2, \dots, x_n\} \setminus \{x_{\min}\}. \end{aligned} \tag{9}$$

where $x_1, x_2, \dots, x_n$ are n neighboring pixels values.

Some exisiting "adaptive embedding" methods [26,27] consider non-overlapping 2×2 pixels as an embedding unit, and the difference value of four-pixel values can be regarded as the above construction in the case of $n = 4$. It would be utilized in the proposed "Ours-AE1" method in our experiments.

$$\begin{aligned} &G(x_1, x_2, x_3, x_4) = \sum_{x_i \in S} (x_i - x_{\min})/3, \\ &x_{\min} = \min\{x_1, x_2, x_3, x_4\},\ S = \{x_1, x_2, x_3, x_4\} \setminus \{x_{\min}\}. \end{aligned} \tag{10}$$

where x_1, x_2, x_3 and x_4 are four neighboring pixels values.

Furthermore, the pixel correlation function G can be designed according to the local complexity. Divide the digital image into non-overlapping n pixels, and the function G is defined based on the variance of pixel values.

$$G(x_1, x_2, \dots, x_n) = \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 / n}, \qquad \bar{x} = \sum_{i=1}^{n} x_i / n. \tag{11}$$

where $x_1, x_2, \dots, x_n$ are n neighboring pixels values.

If $n = 4$, the pixel correlation function G is as follows. It would be used in the proposed "Ours-AE2" method in our experiments.

$$G(x_1, x_2, x_3, x_4) = \sqrt{\sum_{i=1}^{4} (x_i - \bar{x})^2 / 4}, \qquad \bar{x} = \sum_{i=1}^{4} x_i / 4. \tag{12}$$

where x_1, x_2, x_3 and x_4 are four neighboring pixels values.

3.2. The proposed "adaptive embedding" framework

In this subsection, an "adaptive embedding" framework is proposed based on the steganographic embedding function F and the pixel correlation function G. The detailed embedding process and extraction process are given as follows.

3.2.1. Embedding process

We first transform the binary secret bit stream into decimal value m, and then the proposed framework will conceal secret data m into the cover image. The cover image is partitioned into non-overlapping n-pixel blocks, and their corresponding gray values are $\mathbf{p} = (p_1, p_2, \dots, p_n)$. The range of pixel correlation function value $G(\mathbf{p})$ is partitioned into two different levels. The threshold value T is predefined by users. Low level will use lower parameter values k_l and $b_l = 1 + \sum_{i=1}^{n} \binom{n}{i} \binom{k_l}{i} \times 2^i$, while high level uses higher parameter values k_h and $b_h = 1 + \sum_{i=1}^{n} \binom{n}{i} \binom{k_h}{i} \times 2^i$. For each n-pixel block, the detailed embedding steps are as follows.

Step 1: Calculate the pixel correlation function value $G(\mathbf{p})$. If $G(\mathbf{p}) < T$, it belongs to lower level, $k_1 = k_l$ and $b_1 = b_l$ (i.e., the block belongs to a smooth area). Otherwise, it belongs to higher level, $k_1 = k_h$ and $b_1 = b_h$ (i.e., the block belongs to an edge area). Calculate the steganographic embedding function value $F(\mathbf{p}) = \mathbf{A}\mathbf{p}^T \bmod b_1$.

Step 2: Calculate $s = \bmod(m, b_1)$ and $d = s - F(\mathbf{p}) \bmod b_1$. If $d = 0$, no modification for the n pixels is needed, and the secret digit s has been embedded. Replace m by $(m - s)/b_1$, and modify the pixel values, and then run the next n-pixel block. Otherwise, go to the next step.

Step 3: Since the function F is one-to-one function, we can find the unique vector $\mathbf{x}$ such that $F(\mathbf{x}) = d$ and $\|\mathbf{x}\|_1 \le k$.

Step 4: Calculate $\mathbf{p}' = (p'_1, p'_2, \dots, p'_n)$ by $\mathbf{p}' = \mathbf{p} + \mathbf{x}$, and adjust $\mathbf{p}' = (p'_1, p'_2, \dots, p'_n)$ to $\mathbf{p}'' = (p''_1, p''_2, \dots, p''_n)$ when one stego-pixel value has the overflow or underflow problem. If $p'_i > 255$, then $p''_i = p'_i - b_1$. If $p'_i < 0$, then $p''_i = p'_i + b_1$. If $0 \le p'_i \le 255$, then $p''_i = p'_i$.

Step 5: Calculate the function value $G(\mathbf{p}'')$. If $G(\mathbf{p})$ and $G(\mathbf{p}'')$ belong the same level, the secret digit s has been embedded successfully. Replace $\mathbf{p}$ and m by $\mathbf{p}''$ and $(m - s)/b_1$, and modify the pixel values, and then go to step 7. If $G(\mathbf{p})$ and $G(\mathbf{p}'')$ belong different levels, the receiver cannot extract the secret digit s exactly, and we should go to the next step.

Step 6: This step is called "adjusting step". There are two adjusting cases.

Case 1: Adjust the pixels to guarantee the same level that the pixel correlation function values belong to before and after embedding. Search $\mathbf{x}^{(1)} = (x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}) \in \mathbb{Z}^n$ satisfying the following four conditions. (1) $G(\mathbf{p})$ and $G(\mathbf{x}^{(1)} + \mathbf{p})$ belong to the same level. (2) $F(\mathbf{x}^{(1)} + \mathbf{p}) = s$. (3) $0 \le x_i^{(1)} + p_i \le 255$. (4) The value of $\|\mathbf{x}_1^{(1)}\|_2$ is minimized.

Case 2: Modify the secret digit to be embedded in this n-pixel block. If $b_1 = b_l$, then $b_2 = b_h$. Otherwise $b_2 = b_l$. Calculate $s' = \bmod(m, b_2)$. The block will be embedded the secret digit s'. Search $\mathbf{x}^{(2)} = (x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}) \in \mathbb{Z}^n$ satisfying the following four conditions. (1) $G(\mathbf{p})$ and $G(\mathbf{x}^{(2)} + \mathbf{p})$ belong to different levels. (2) $F(\mathbf{x}^{(2)} + \mathbf{p}) = s'$. (3) $0 \le x_i^{(2)} + p_i \le 255$. (4) The value of $\|\mathbf{x}^{(2)}\|_2$ is minimized.

If $\frac{\log_2 b_1}{\|\mathbf{x}^{(1)}\|_2} > \frac{\log_2 b_2}{\|\mathbf{x}^{(2)}\|_2}$, *Case* 1 provides larger payload with smaller embedding distortion, and then replace $\mathbf{p}$ and m by $\mathbf{p} + \mathbf{x}^{(1)}$ and $(m - s)/b_1$. Otherwise, *Case* 2 provides larger payload with smaller embedding distortion, and then replace $\mathbf{p}$ and m by $\mathbf{p} + \mathbf{x}^{(2)}$ and $(m - s')/b_2$.

Step 7: Repeat the above steps until $m = 0$, and then the stego image is obtained.

3.2.2. Extraction process

In the extraction process, the secret data m can be extracted without the original image. Partition the stego image into n-pixel blocks, which is identical with the embedding procedure.

Step 1: For each n-pixel block, their corresponding gray values are $\mathbf{p}'' = (p''_1, p''_2, \dots, p''_n)$, and calculate the pixel correlation function value $G(\mathbf{p}'')$. Use the threshold value T to find out the level which $G(\mathbf{p}'')$ belongs to. If $G(\mathbf{p}'') < T$, then $k = k_l$ and $b = b_l$. Otherwise, $k = k_h$ and $b = b_h$.

Step 2: Calculate the function value $F(\mathbf{p}'') = \mathbf{A}\mathbf{p}''^T \bmod b$. Extract the embedded digit $s = F(\mathbf{p}'')$.

Step 3: Repeat steps 1–2 until all the embedded digits in each n-pixel block are extracted.

Step 4: Assume that all the extracted digits are $(s^{(1)}, s^{(2)}, \dots, s^{(l)})$, and their corresponding bases (modulus) are $(b^{(1)}, b^{(2)}, \dots, b^{(l)})$. Calculate the secret data m using the equation

$$m = \sum_{i=2}^{l} \left(s^{(i)} \times \prod_{j=1}^{i-1} b^{(j)} \right) + s^{(1)} \tag{13}$$

3.3. A simple example

In this subsection, we use a simple example to illustrate the proposed framework. Suppose a cover image is composed of twelve pixels, and their corresponding gray values are (170,168,175,170,175,176,166,172, 167,169,169,170). The secret bits to be embedded are 10000100111100_2, i.e., the decimal value $m = 8508$. The parameters of the proposed "adaptive embedding" framework are $T = 4$, $k_l = 2$, $k_h = 3$, $n = 4$, and the pixel correlation function G is defined as Eq. (10). The detailed steps are shown in Fig. 2.

To embed secret data, the cover image is partitioned into three blocks with four pixel $(170, 168, 175, 170)$, $(175, 176, 166, 172)$ and $(167, 169, 169, 170)$. For the first block, $G(\mathbf{p}) = G(170, 168, 175, 170) = 11/3 < T$, it belongs to lower level, $k_1 = 2$ and $b_1 = 41$. Calculate the function value $F(170, 168, 175, 170) = 26$, $s^{(1)} = \bmod(8508, 41) = 21$, and then $d = 36$. The suitable vector $\mathbf{x} = (0, -1, 0, 0)$ satisfies $F(\mathbf{x}) = 36$, so $\mathbf{p}'' = (170, 167, 175, 170)$ and $G(\mathbf{p}'') = 14/3 > T$. Since $G(\mathbf{p})$ and $G(\mathbf{p}'')$ belong different levels, the receiver cannot extract the secret digit $s^{(1)}$ exactly, and we have to execute the "adjusting step". $\mathbf{x}^{(1)} = (-1, 0, 0, -2)$ in *Case* 1 and $\mathbf{x}^{(2)} = (1, 0, 0, -3)$ in *Case* 2. $\frac{\log_2 41}{\|\mathbf{x}_1\|_2} > \frac{\log_2 129}{\|\mathbf{x}_2\|_2}$, *Case* 1 provides larger payload with smaller embedding distortion. Thus, replace $(170, 168, 175, 170)$ and $m = 8508$ by $(169, 168, 175, 168)$ and $m = (8508 - 21)/41 = 207$. For the second block, $G(\mathbf{p}) = G(175, 176, 166, 172) = 25/3 > T$, it belongs to higher level, $k_1 = 3$ and $b_1 = 129$. Calculate $F(175, 176, 166, 172) = 38$, $s^{(2)} = \bmod(207, 129) = 78$ and $d = 40$. The suitable vector $\mathbf{x} = (-2, -1, 1, 0)$ such that $F(\mathbf{x}) = 40$, so $\mathbf{p}'' = (173, 175, 167, 172)$. Since $G(\mathbf{p})$ and $G(\mathbf{p}'')$ belong the same level, the secret digit $s^{(2)}$ has been embedded successfully. Replace $(175, 176, 166, 172)$ and $m = 207$ by $(173, 175, 167, 172)$ and $m = (207 - 78)/129 = 1$. For the third block, $G(\mathbf{p}) = G(167, 169, 169, 170) = 7/3 < T$, it belongs to lower level, $k_1 = 2$ and $b_1 = 41$. Calculate $F(167, 169, 169, 170) = 1$, $s^{(3)} = \bmod(1, 41) = 1$

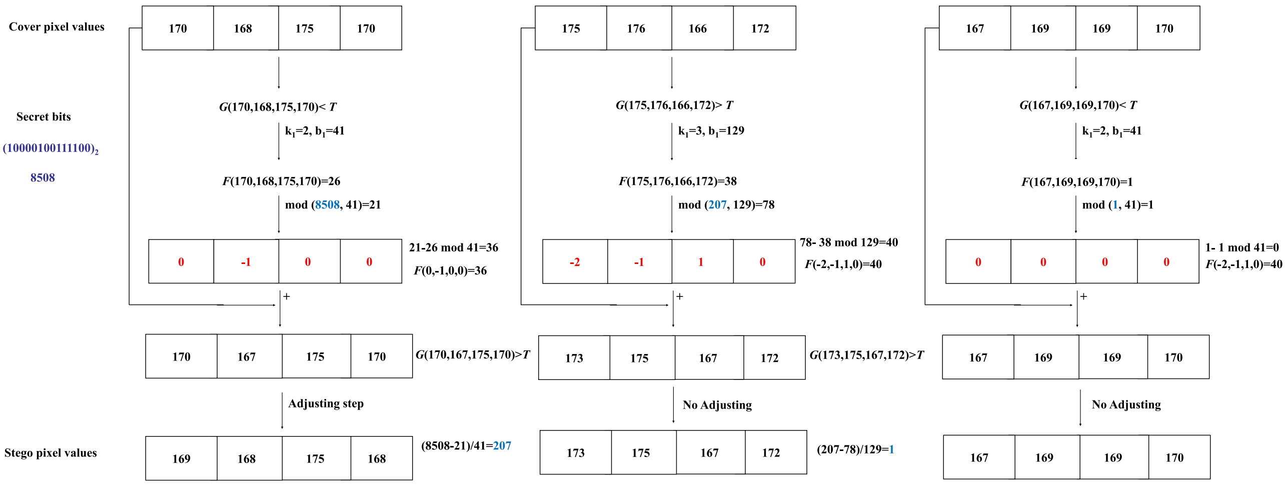

Fig. 2. A simple example of the proposed "Adaptive Embedding" framework.

and $d = 0$. No modification for these pixels is needed, and the secret digit $s^{(3)}$ has been embedded successfully. The stego image is obtained.

The secret data can be extracted without the original image. Partition the stego image into three blocks. For each four-pixel block, calculate the pixel correlation function value $G(169, 168, 175, 168) = 8/3$, $G(173, 175, 167, 172) = 19/3$ and $G(167, 169, 169, 170) = 7/3$. Use the threshold value $T = 4$ to find out the level which G belongs to, $b^{(1)} = 41$, $b^{(2)} = 129$ and $b^{(3)} = 41$. Extract the embedded digit $s^{(1)} = F(169, 168, 175, 168) = 21$, $s^{(2)} = F(173, 175, 167, 172) = 78$ and $s^{(3)} = F(167, 169, 169, 170) = 1$. According to Eq. (13), $m = 21 + 78 \times 41 + 1 \times 41 \times 129 = 8508$, so we can obtain the embedded secret bits 10000100111100_2.

3.4. Analysis

The adjusting step of the proposed framework is the essential technique, ensuring extracting the secret data successfully. It is different from that of other "adaptive embedding" methods. In Refs. [11,15,17,18,26,27], the pixels are only adjusted to guarantee the same level that the pixel correlation function values belong to before and after embedding (*Case* 1 in Step 6). However, there exists another adjusting case (*Case* 2 in Step 6) in the proposed framework. The secret data m can be dynamically expressed as different digits s in different modulus b, so the digits to be embedded in each n-pixel block can be modified.

According to the detailed embedding procedure, we know that the secret data m is recalculated by an iterative process $m = (m - s)/b$, and the secret data m is expressed as digits $(s^{(1)}, s^{(2)}, \dots, s^{(l)})$ in different bases $(b^{(1)}, b^{(2)}, \dots, b^{(l)})$, $s^{(1)} = \mathrm{mod}\ (m, b^{(1)})$, $s^{(i)} = \mathrm{mod}\ (\frac{m-s^{i-1}}{b^{(i-1)}}, b^{(i)})(i \geq 2)$, by recalculating an iterative process $m = (m-s)/b$. Thus, we can execute the reverse operation $\sum_{i=2}^{l}(s^{(i)} \times \prod_{j=1}^{i-1} b^{(j)}) + s^{(1)}$ and then obtain the secret data m reversely, i.e., the receiver can exactly extract the secret data.

Compared with the proposed "Uniform Embedding" framework in Section 2.2, Step 6 ("adjusting step") in the proposed "Adaptive Embedding" framework might introduce extra computational costs. Fortunately, since the maximum searching range of each pixel is 2^8 in the adjusting step, the computational cost is no more than $2^8 \times n$ even if the exhaustive searching is used. Thus, the computation complexity of Step 6 is $O(n)$. The total computation complexity of the proposed framework is still $O(nk^2)$, and then the proposed "Adaptive Embedding" framework could be solved in polynomial time when the parameter k is small. In fact, the time cost for each image is less than 0.5 s. Specifically, for the classic image "Lena", it will cost only 0.46 s.

3.5. Discussions

The following constructions of the functions F and G make it possible to derive some previous steganographic methods from the proposed framework.

AdEMD [11]: AdEMD method executes two-pixel pair at a time, and the coefficients of the function F is changed to $a_1 = 1$, $a_2 = 2k_1 + 1$ and $b = 1$. AdEMD restricts the modification of each pixel, so the constraints are altered to $|x_1| \leq k_1$ and $|x_2| \leq k_2$. The function G is defined as Eq. (8). Furthermore, AdEMD only adjusts the pixels to guarantee the same level that the pixel correlation function values belong to before and after embedding, i.e., only *Case* 1 is included in the adjusting step of embedding procedure.

$$\begin{gathered} F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b, \\ x_i \equiv p_i \bmod (2k_i + 1),\ a_i = \prod_{j=0}^{i-1}(2k_j + 1),\ k_0 = 0,\ b = 1, \\ G(p_1, p_2) = |p_1 - p_2| \end{gathered} \tag{14}$$

where p_1 and p_2 are two neighboring pixels values.

DE-PVD [19]: DE-PVD method executes two-pixel pair at a time, and the coefficients of the function F is changed to $a_1 = 1$, $a_2 = 2k + 1$ and $b = 1 + 2k + 2k^2$. The function G is defined as Eq. (8).

$$\begin{gathered} F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b, \\ \|\mathbf{x}\|_1 \leq k,\ a_i = (2k+1)^{i-1},\ b = 1 + 2k + 2k^2, \\ G(x_1, x_2) = |x_1 - x_2| \end{gathered} \tag{15}$$

where x_1 and x_2 are two neighboring pixels values.

EMD-ADFPV [27]: EMD-ADFPV method executes four-pixel block at a time, and the coefficients of the function F is changed to $a_1 = 1$, $a_2 = 2k_1 + 1$, $a_3 = (2k_1 + 1)(2k_2 + 1)$, $a_4 = (2k_1 + 1)(2k_2 + 1)(2k_3 + 1)$ and $b = 1$. The function G is defined as Eq. (10). EMD-ADFPV restricts the modification of each pixel $|x_i| \leq k_i$. Only *Case* 1 is included in the adjusting step of embedding procedure.

$$\begin{gathered} F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b, \\ x_i \equiv p_i \bmod (2k_i + 1),\ a_i = \prod_{j=0}^{i-1}(2k_j + 1),\ k_0 = 0,\ b = 1, \\ G(p_1, p_2, p_3, p_4) = \sum_{p_i \in S}(p_i - p_{\min})/3, \\ p_{\min} = \min\{p_1, p_2, p_3, p_4\},\ S = \{p_1, p_2, p_3, p_4\} \setminus \{p_{\min}\} \end{gathered} \tag{16}$$

where p_1, p_2, p_3 and p_4 are four neighboring pixels values.

4. Experimental results

In this section, several experimental results are given to demonstrate the effectiveness of our proposed "uniform embedding" and "adaptive embedding" methods.

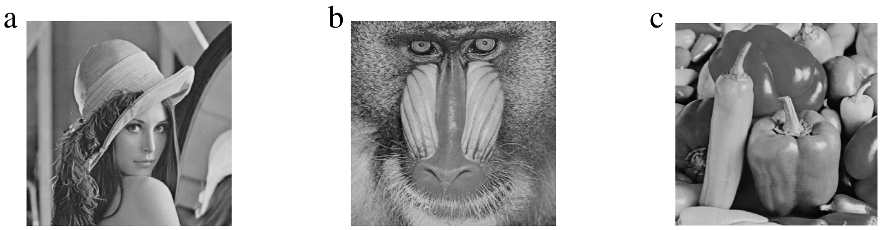

Fig. 3. Three cover images (a) Lena (b) Baboon (c) Peppers.

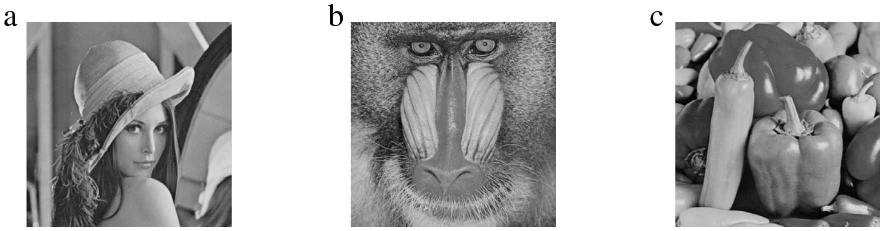

Fig. 4. Three stego images created by the proposed “Ours-UE” method (a) Lena (60 000 bits, PSNR = 44.72 dB, SSIM = 0.9792) (b) Baboon (60 000 bits, PSNR = 44.72 dB, SSIM = 0.9816) (c) Peppers (60 000 bits, PSNR = 44.72 dB, SSIM = 0.9937).

4.1. General setup

The peak signal to noise ratio (PSNR) and structure similarity index (SSIM) [39] are utilized to evaluate the quality of stego images. For a $M \times N$ grayscale image, the PSNR value is

$$PSNR = 10 \times \log_{10} \frac{255 \times 255 \times M \times N}{\sum_{i=1}^{M} \sum_{j=1}^{N} (p_{i,j} - q_{i,j})^2} \ (dB) \tag{17}$$

where $p_{i,j}$ and $q_{i,j}$ denote the pixel values in row i and column j of the cover image and stego image, respectively. SSIM is a full reference metric for measuring image similarity by considering the human eye perception.

The image steganographic security is analyzed by employing modern universal steganalysis techniques, and the performance is evaluated using the detection error rate P_{Err}, which is an unbiased estimate of the minimal total testing error under equal priors.

There is a trade-off among embedding capacity, visual imperceptibility and attack-resistance. It would sacrifice embedding capacity a little for acquiring better image quality and steganographic security, and vice versa. Therefore, the comparisons of different image steganographic methods should be based on the same criterion: a steganographic method is preferred if it can provide better imperceptibility and undetectability when concealing the same embedding capacity. Therefore, we will conduct the comparative experiments based on the same embedding capacity. All the parameter settings ensure that the corresponding methods can provide enough capacity while maintaining the smallest image distortions and highest statistical undetectability. In DE [8] method, the parameter $k = 2$. In APPM [9] method, the parameters $B = 9$ and $C = 3$. In AdEMD [11] method, $D_{lh} = 15$, $\langle lk_1, lk_2 \rangle$ and $\langle hk_1, hk_2 \rangle$ are set $\langle 2, 3 \rangle$ and $\langle 4, 5 \rangle$. In EA-PVD [17] method, $D_{12} = 7$, $l - h$ is set $2 - 3$. In DE-PVD [19] method, $T_0 = 4, k_l = 2, k_h = 3$. We construct the special image steganographic methods by using the proposed frameworks in the case of four neighboring pixel as a processing unit. We set $k = 2$ in the proposed “uniform embedding” method (abbreviated as “Ours-UE”). In the proposed “adaptive embedding” method, $T = 2, k_l = 2, k_h = 3$, the pixel correlation function G is respectively defined as Eqs. (10) and (12) (abbreviated as “Ours-AE1” and “Ours-AE2”).

4.2. Performances for classic images

Table 1
Comparison of PSNR values of “Uniform Embedding” methods based on classic images.

	Ours-UE		DE [8]		APPM [9]	
	350 000	600 000	350 000	600 000	350 000	600 000
Elaine	51.05	44.71	49.22	43.48	50.64	44.32
Lena	51.04	44.72	49.22	43.47	50.64	44.32
Baboon	51.05	44.72	49.21	43.48	50.64	44.32
Peppers	51.05	44.71	49.22	43.48	50.64	44.32
Toys	51.05	44.72	49.22	43.48	50.64	44.32
Girl	51.05	44.72	49.22	43.47	50.64	44.32
Gold	51.05	44.71	49.22	43.48	50.64	44.32
Barb	51.04	44.72	49.21	43.48	50.64	44.32
Zelda	51.05	44.72	49.22	43.47	50.64	44.32
Tiffany	51.05	44.72	49.22	43.48	50.64	44.32
Average	**51.05**	**44.72**	**49.22**	**43.48**	**50.64**	**44.32**

Table 2
Comparison of SSIM values of “Uniform Embedding” methods based on classic images.

	Ours-UE		DE [8]		APPM [9]	
	350 000	600 000	350 000	600 000	350 000	600 000
Elaine	0.9969	0.9867	0.9953	0.9828	0.9857	0.9843
Lena	0.9951	0.9792	0.9926	0.9726	0.9775	0.9757
Baboon	0.9957	0.9816	0.9931	0.9755	0.9800	0.9782
Peppers	0.9985	0.9937	0.9979	0.9917	0.9933	0.9924
Toys	0.9956	0.9813	0.9934	0.9753	0.9800	0.9780
Girl	0.9955	0.9806	0.9926	0.9736	0.9786	0.9772
Gold	0.9965	0.9853	0.9949	0.9809	0.9843	0.9823
Barb	0.9969	0.9869	0.9953	0.9832	0.9860	0.9844
Zelda	0.9973	0.9885	0.9957	0.9852	0.9878	0.9863
Tiffany	0.9954	0.9803	0.9931	0.9744	0.9787	0.9770
Average	**0.9963**	**0.9844**	**0.9944**	**0.9795**	**0.9832**	**0.9816**

Ten grayscale images with size of 512×512 are used in the experiments as cover images, and three of them are shown in Fig. 3. A series of pseudo-random numbers as the secret bit stream are embedded into the cover images. Stego images created by the proposed “Ours-UE”, “Ours-AE1” and “Ours-AE2” method are shown in Figs. 4–6. It is shown that the embedding distortions are imperceptible to human vision, and the stego images are visually indistinguishable from the cover images.

4.3. Comparisons of performances

The comparison results of PSNR and SSIM values among Ours-UE, DE [8] and APPM [9] are given in Tables 1–2, using ten classic images at low embedding capacity (350 000 bits) and high embedding capacity (600 000 bits). The experimental comparisons among Ours-AE1, Ours-AE2, AdEMD [11], EA-PVD [17] and DE-PVD [19] are shown

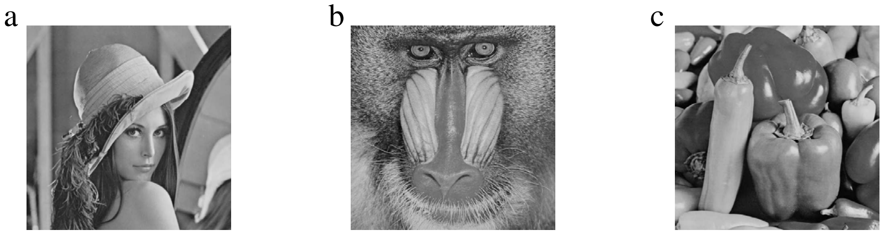

Fig. 5. Three stego images created by the proposed “Ours-AE1” method (a) Lena (60 000 bits, PSNR = 44.85 dB, SSIM = 0.9826) (b) Baboon (60 000 bits, PSNR = 44.85 dB, SSIM = 0.9846) (c) Peppers (60 000 bits, PSNR = 44.85 dB, SSIM = 0.9939).

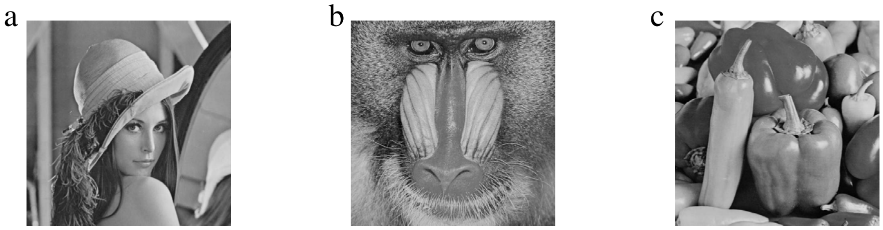

Fig. 6. Three stego images created by the proposed “Ours-AE2” method (a) Lena (60 000 bits, PSNR = 47.54 dB, SSIM = 0.9903) (b) Baboon (60 000 bits, PSNR = 46.67 dB, SSIM = 0.9914) (c) Peppers (60 000 bits, PSNR = 47.71 dB, SSIM = 0.9964).

Table 3
Comparison of PSNR values of “Adaptive Embedding” methods based on classic images.

	Ours-AE1		Ours-AE2		AdEMD [11]		DE-PVD [19]		EA-PVD [17]	
	350 000	600 000	350 000	600 000	350 000	600 000	350 000	600 000	350 000	600 000
Elaine	50.67	44.85	50.74	46.93	44.94	40.87	47.54	43.36	44.73	39.99
Lena	50.67	44.85	50.95	47.54	45.57	41.69	48.06	43.79	46.11	40.81
Baboon	50.67	44.85	50.68	46.67	43.99	39.30	47.46	43.37	44.55	39.01
Peppers	50.67	44.85	50.99	47.71	45.49	42.24	48.09	43.81	46.15	40.96
Toys	50.67	44.85	51.02	47.86	45.66	42.06	48.04	43.67	46.31	40.95
Girl	50.67	44.85	50.91	47.41	45.38	41.79	47.87	43.65	45.43	40.68
Gold	50.67	44.85	50.89	47.40	45.55	41.54	47.88	43.61	45.67	40.45
Barb	50.67	44.85	50.84	47.03	44.86	39.63	47.73	43.52	45.30	39.38
Zelda	50.67	44.85	50.91	47.71	45.62	42.69	47.95	43.71	45.97	41.15
Tiffany	50.67	44.85	50.98	47.69	45.48	42.05	48.01	43.78	45.92	40.73
Average	**50.76**	**45.01**	**50.89**	**47.39**	**45.25**	**41.39**	**47.86**	**43.63**	**45.61**	**40.41**

Table 4
Comparison of SSIM values of “Adaptive Embedding” methods based on classic images.

	Ours-AE1		Ours-AE2		AdEMD [11]		DE-PVD [19]		EA-PVD [17]	
	350 000	600 000	350 000	600 000	350 000	600 000	350 000	600 000	350 000	600 000
Elaine	0.9967	0.9873	0.9968	0.9925	0.9885	0.9752	0.9934	0.9826	0.9862	0.9598
Lena	0.9949	0.9826	0.9952	0.9903	0.9845	0.9712	0.9910	0.9755	0.9851	0.9483
Baboon	0.9956	0.9846	0.9958	0.9914	0.9852	0.9756	0.9920	0.9790	0.9872	0.9535
Peppers	0.9984	0.9939	0.9985	0.9964	0.9957	0.9885	0.9974	0.9920	0.9952	0.9828
Toys	0.9954	0.9841	0.9957	0.9914	0.9868	0.9759	0.9923	0.9783	0.9880	0.9545
Girl	0.9950	0.9838	0.9954	0.9914	0.9841	0.9743	0.9908	0.9762	0.9858	0.9501
Gold	0.9965	0.9871	0.9967	0.9926	0.9889	0.9788	0.9936	0.9824	0.9882	0.9625
Barb	0.9969	0.9884	0.9970	0.9935	0.9894	0.9810	0.9940	0.9840	0.9893	0.9661
Zelda	0.9972	0.9899	0.9974	0.9943	0.9899	0.9824	0.9946	0.9864	0.9908	0.9686
Tiffany	0.9952	0.9830	0.9954	0.9909	0.9865	0.9745	0.9915	0.9762	0.9866	0.9522
Average	**0.9962**	**0.9865**	**0.9964**	**0.9925**	**0.9880**	**0.9777**	**0.9931**	**0.9813**	**0.9882**	**0.9598**

in Tables 3–4. It is observed that the proposed steganographic methods can obtain better image quality when concealing the same embedding capacity.

To further evaluate the performances, three image databases are used in the experiments as follows.

(1) UCID Database [40]: 1338 uncompressed images with size of 384 × 512 or 512 × 384.

(2) McGill Database [41]: 1034 uncompressed images with size of 768 × 576 or 576 × 768.

(3) Ground Truth Database [42]: 1333 uncompressed images with size of 756 × 504 or 504 × 756.

3705 test images include (but not limited to) landscapes, plants, animals, people and buildings, and have been converted into grayscale images. Stego images are created with payload 0.1 bpp (bit per pixel), 0.2 bpp, 0.3 bpp, 0.4 bpp and 0.5 bpp using these image steganographic methods. The comparison results of PSNR and SSIM values are shown in Tables 5–8, respectively. It is shown that the proposed steganographic methods can provide better visual imperceptibility with the same payload. Specially, the PSNR values of “Uniform embedding” are increased by 0.24–2.89 dB, and those of “Adaptive embedding” are improved by 2.71–7.69 dB.

4.4. Comparisons of steganographic security

The steganographic security is analyzed by employing the following three universal steganalysis techniques. The image features are

Table 5
Comparison of PSNR values of "Uniform Embedding" methods based on the databases.

	Ours-UE	DE [8]	APPM [9]
0.1 bpp	62.21	59.37	61.97
0.2 bpp	59.21	56.37	58.95
0.3 bpp	57.45	54.63	57.18
0.4 bpp	56.21	53.39	55.93
0.5 bpp	55.32	52.43	54.99

Table 6
Comparison of SSIM values of "Uniform Embedding" methods based on the databases.

	Ours-UE	DE [8]	APPM [9]
0.1 bpp	0.9996	0.9990	0.9996
0.2 bpp	0.9993	0.9981	0.9993
0.3 bpp	0.9990	0.9973	0.9989
0.4 bpp	0.9987	0.9965	0.9986
0.5 bpp	0.9984	0.9958	0.9983

extracted from the cover images and stego images, respectively. The steganalyzer is constructed by using soft-margin supporting vector machine (SVM) [43] with the Gaussian kernel. The values of the penalization parameter C and the kernel parameter γ are searched by the genetic algorithms, and then the lowest detection error rate P_{Err} is obtained.

(1) Li [44]: A set of universal steganalytic 110-dimensional features are extracted from the normalized histograms of the local linear transform coefficients of an image.

(2) SPAM [45]: The local dependences between differences of neighboring cover elements are modeled as a Markov chain, and the transition probability matrix is used as the 686-dimensional features for steganalysis.

(3) SRM [46]: The 34,671-dimensional features consisting of many submodels. The submodels consider various types of relationships among neighboring samples of noise residuals obtained by linear and non-linear filters.

The experimental results are shown in Tables 9 and 10, respectively. The detection error rates P_{Err} obtained by the propose image steganographic methods are higher than those of other existing methods, indicating that the propose image steganographic methods are less detectable when concealing the same embedding payload. For example, with the payload of 0.5 bpp, the detection error rates of DE [8] and APPM [9] against SPAM steganalysis are 18.75 and 19.27. The detection error rate of the proposed "Uniform Embedding" method is 19.96, which is improved by 6.5% and 3.5%.

4.5. Analysis

Compared with prior works, the proposed image steganographic methods can provide better imperceptibility and undetectability when concealing the same embedding capacity. The proposed methods fully take advantage of the steganographic embedding function F and pixel correlation function G, and the detailed analysis is given as follows.

The proposed steganographic embedding function F restricts the maximal modification of n-pixel block, rather than the maximal modification of each pixel. The non-overlapping n-pixel block can be synchronously modified and fully considered, which would contribute to the embedding distortion minimization. Furthermore, the constructions of the function coefficients not only guarantee the exact data extraction, but also increase the embedding capacity.

The proposed pixel correlation function G utilizes the average differences and variances to discriminate the image smoothness. The features of edge are considered sufficiently and the pixels in edge areas could tolerate much more changes without making perceptible and detectable distortion. In addition, the dynamical adjustment could modify the embedded digit. The secret data are dynamically expressed as different digits in multiple-base, and the digits to be embedded in each pixel block can be flexibly adjusted, so as to minimize the embedding distortion.

The experiment results show that, if the embedding capacity is low, Ours-UE, Ours-AE1, Ours-AE2 could obtain good performance of image quality and anti-steganalysis. However, if more payloads would be embedded, Ours-AE1 and Ours-AE2 could obtain higher PSNR values, SSIM values and detection error rates than Ours-UE, when concealing the same capacity. In general, the "adaptive embedding" methods could obtain better performances because it considers the characteristics of human visual system and image textures. Thus, the proposed "adaptive embedding" framework can be considered as more suitable for practical data hiding, so as to acquire better visual imperceptibility and steganographic security.

5. General-purpose strategies

The presented frameworks can be considered as the general and flexible strategies to acquire new data hiding methods. For the given steganographic embedding function F and pixel correlation function G, the proposed frameworks provide efficient tools for constructing practical image steganographic methods, i.e., we can directly obtain new image steganographic methods according to the proposed embedding and extracting procedures. Therefore, the design of image steganography methods are converted into the construction of two functions F and G.

The general form of the steganographic embedding function F would be $F(\mathbf{x}) = \mathbf{A}\mathbf{x}^T \bmod b$. The definition domain Ω and coefficients $\mathbf{A}$ and b are important. Their general forms are as follows.

$$\begin{cases} \Omega : \{\mathbf{x} \mid \|\mathbf{x}\|_l \le k\}, \quad l \in \{1, 2, \infty\} \\ \mathbf{A} = (f_1(k,1), \ldots, f_1(k,i), \ldots, f_1(k,n)), \quad b = f_2(k). \end{cases} \tag{18}$$

The design of F is determined by the construction of the subfunctions f_1 (a function of two independent variables k and i), and f_2 (a function

Table 7
Comparison of PSNR values of "Adaptive Embedding" methods based on the databases.

	Ours-AE1	Ours-AE2	AdEMD [11]	DE-PVD [19]	EA-PVD [17]
0.1 bpp	62.23	62.41	56.35	59.59	55.21
0.2 bpp	59.21	59.38	53.31	56.56	51.99
0.3 bpp	57.30	57.47	51.53	54.76	50.09
0.4 bpp	56.04	56.28	50.26	53.47	48.72
0.5 bpp	55.18	55.33	49.27	52.48	47.64

Table 8
Comparison of SSIM values of "Adaptive Embedding" methods based on the databases.

	Ours-AE1	Ours-AE2	AdEMD [11]	DE-PVD [19]	EA-PVD [17]
0.1 bpp	0.9997	0.9997	0.9988	0.9994	0.9990
0.2 bpp	0.9994	0.9994	0.9977	0.9989	0.9980
0.3 bpp	0.9991	0.9991	0.9966	0.9983	0.9970
0.4 bpp	0.9988	0.9989	0.9956	0.9978	0.9961
0.5 bpp	0.9986	0.9986	0.9946	0.9974	0.9952

Table 9
Comparison of detection error rates of “Uniform Embedding” methods.

	Ours-UE			DE [8]			APPM [9]		
	Li	SPAM	SRM	Li	SPAM	SRM	Li	SPAM	SRM
0.1 bpp	24.37	23.93	28.90	24.13	23.16	24.02	23.17	22.90	25.06
0.2 bpp	23.60	23.09	22.54	23.47	22.41	18.56	22.51	22.41	19.56
0.3 bpp	21.58	21.36	19.12	21.39	21.36	15.69	21.52	21.27	16.21
0.4 bpp	21.36	20.93	16.85	20.77	20.16	13.55	20.76	20.09	14.04
0.5 bpp	20.64	19.96	15.04	20.09	18.75	11.59	20.33	19.27	12.33

Table 10
Comparison of detection error rates of “Adaptive Embedding” methods.

	Ours-AE1			Ours-AE2			AdEMD [11]			DE-PVD [19]			EA-PVD [17]		
	Li	SPAM	SRM	Li	SPAM	SRM	Li	SPAM	SRM	Li	SPAM	SRM	Li	SPAM	SRM
0.1 bpp	24.60	24.18	30.44	24.62	24.40	30.62	23.01	23.57	18.05	23.58	22.68	10.55	23.83	23.86	24.03
0.2 bpp	23.80	23.48	24.35	23.96	23.66	24.58	22.36	22.69	15.83	22.66	22.02	5.98	22.58	22.97	18.07
0.3 bpp	22.28	21.86	20.57	22.60	21.98	20.78	21.40	21.40	13.30	21.86	21.06	4.10	20.87	20.95	13.26
0.4 bpp	21.72	21.40	18.02	21.90	21.44	18.28	20.68	20.57	11.32	20.98	20.41	2.94	18.40	19.52	10.12
0.5 bpp	21.11	20.71	16.04	21.18	20.70	16.12	20.08	19.77	9.59	20.48	19.81	2.21	15.59	18.54	7.13

of one variable k). Furthermore, the steganographic embedding function F should be a bijective mapping.

$$\begin{cases} F(\mathbf{x}_1) \neq F(\mathbf{x}_2) \quad \text{if } \mathbf{x}_1 \neq \mathbf{x}_2, \\ \forall y \in \mathbb{N}, \exists\, \mathbf{x} \in \Omega, \quad F(\mathbf{x}) = y. \end{cases} \tag{19}$$

The embedding rate (the number of embedded bits in each pixel) and embedding efficiency (a ratio between the number of embedded bits and the embedding distortion D) are used for performance evaluation. The embedding rate is $\lfloor \log_2 b \rfloor / n = \lfloor \log_2 f_2(k) \rfloor / n$, and the embedding efficiency is $\lfloor \log_2 b \rfloor / D = \lfloor \log_2 f_2(k) \rfloor / D$. The embedding distortion D is determined by the domain Ω.

The pixel correlation function G is limited as average differences and variances of pixel values in this paper. It is worth investigating other function forms for describing image textures fully, such as image entropy and gray-level co-occurrence matrix. That is a part of our future work.

6. Conclusions

In this paper, the steganographic embedding function and pixel correlation function are firstly introduced, and then two efficient steganographic frameworks for “uniform embedding” and “adaptive embedding” based on them are proposed. Some well-known image steganographic embedding methods can be obtained from the proposed frameworks. We construct the special data hiding methods in the case of four neighboring pixel as a processing unit. Experimental results demonstrate that the new image steganographic methods not only improve the embedding capacity and stego image quality, but also achieve better statistical undetectability compared with the prior works. The presented work could be regarded as the off-the-shelf methodologies that create the efficient image steganographic methods.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant Nos. 61402162, 61572182, 61370225, 61472131, 61272546), National High Technology Research and Development Program of China (Grant No. 2015AA016003), Hunan Provincial Natural Science Foundation of China (Grant No. 2017JJ3040), Science and Technology Key Projects of Hunan Province (Grant Nos. 2015TP1004, 2016JC2012). Dr. Liao is also a visiting researcher with Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China.

References

[1] R.J. Anderson, F.A.P. Petitcolas, On the limits of steganography, IEEE J. Sel. Areas Commun. 16 (1998) 474–481.

[2] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding-a survey, Proc. IEEE Spec. Issue Prot. Multimedia Content 87 (7) (1999) 1062–1078.

[3] H. Wang, S. Wang, Cyber warfare: steganography vs. steganalysis, Commun. ACM 47 (10) (2004) 76–82.

[4] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications, Cambridge University Press, UK, 2009.

[5] C.K. Chan, L.M. Chen, Hiding data in images by simple LSB substitution, Pattern Recognit. 37 (3) (2004) 469–474.

[6] A. Ker, Improved detection of LSB steganography in grayscale images, in: Proceedings of International Information Hiding Workshop, 2004, pp. 97–115.

[7] J. Mielikainen, LSB matching revisited, IEEE Signal Process. Lett. 13 (5) (2006) 285–287.

[8] R.M. Chao, H.C. Wu, C.C. Lee, et al., A novel image data hiding scheme with diamond encoding, EURASIP J. Inf. Secur. (2009) 658047.

[9] W. Hong, T.S. Chen, A novel data embedding method using adaptive pixel pair matching, IEEE Trans. Inf. Forensics Secur. 7 (1) (2012) 176–184.

[10] X. Zhang, S. Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Commun. Lett. 10 (11) (2006) 781–783.

[11] H.M. Sun, C.Y. Weng, C.F. Lee, et al., Anti-forensics with steganographic data embedding in digital images, IEEE J. Sel. Areas Commun. 29 (7) (2011) 1392–1403.

[12] C.C. Chang, H.W. Tseng, A steganographic method for digital images using side match, Pattern Recognit. Lett. 25 (12) (2004) 1431–1437.

[13] X. Zhang, S. Wang, Steganography using multiple-base notational system and human vision sensitivity, IEEE Signal Process. Lett. 12 (1) (2005) 67–70.

[14] D.C. Wu, W.H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognit. Lett. 24 (9–10) (2003) 1613–1626.

[15] H.C. Wu, N.I. Wu, C.S. Tsai, et al., Image steganographic scheme based on pixel-value differencing and LSB replacement methods, IEE Proc. Vis. Image Signal Process. 152 (5) (2005) 611–615.

[16] C.M. Wang, N.I. Wu, C.S. Tsai, A high quality steganography method with pixel-value differencing and modulus function, J. Syst. Softw. 81 (1) (2008) 150–158.

[17] C.H. Yang, C.Y. Weng, S.J. Wang, et al., Adaptive data hiding in edge areas of images with spatial LSB domain systems, IEEE Trans. Inf. Forensics Secur. 3 (3) (2008) 488–497.

[18] W. Luo, F. Huang, J. Huang, Edge adaptive image steganography based on LSB matching revisited, IEEE Trans. Inf. Forensics Secur. 5 (2) (2010) 201–214.

[19] W. Hong, T. Chen, C. Luo, Data embedding using pixel value differencing and diamond encoding with multiple-base notational system, J. Syst. Softw. 85 (5) (2012) 1166–1175.

[20] S. Shen, L. Huang, A data hiding scheme using pixel value differencing and improving exploiting modification directions, Comput. Secur 48 (2015) 131–141.

[21] M. Hussaina, A.W.A. Wahaba, A.T.S. Ho, et al., A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement, Signal Process., Image Commun. 50 (2017) 44–57.

[22] X. Liao, C. Shu, Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels, J. Vis. Commun. Image Represent. 28 (4) (2015) 21–27.

[23] J.C. Liu, M.H. Shih, Generalizations of pixel-value differencing steganography for data hiding in images, Fund. Inform. 83 (3) (2008) 319–335.

[24] C.H. Yang, S.J. Wang, C.Y. Weng, Capacity-raising steganography using multi-pixel differencing and pixel-value shifting operations, Fund. Inform. 98 (2) (2010) 321–336.

[25] X. Liao, Q. Wen, Z. Zhao, et al., A novel steganographic method with four-pixel differencing and modulus function, Fund. Inform. 118 (3) (2012) 281–289.

[26] X. Liao, Q. Wen, J. Zhang, A steganographic method for digital images with four-pixel differencing and modified LSB substitution, J. Vis. Commun. Image Represent. 22 (1) (2011) 1–8.

[27] X. Liao, Q. Wen, J. Zhang, A novel steganographic method with four-pixel differencing and exploiting modification direction, IEICE Trans. Fundam. E95-A (7) (2012) 1189–1192.

[28] C. Balasubramanian, S. Selvakumar, S. Geetha, High payload image steganography with reduced distortion using octonary pixel pairing scheme, Multimedia Tools Appl. 73 (2014) 2223–2245.

[29] J. Chen, A PVD-based data hiding method with histogram preserving using pixel pair matching, Signal Process., Image Commun. 29 (2014) 375–384.

[30] B. Li, S. Tan, M. Wang, et al., Investigation on cost assignment in spatial image steganography, IEEE Trans. Inf. Forensics Secur. 9 (8) (2014) 1264–1277.

[31] T. Pevný, T. Filler, P. Bas, Using high-dimensional image models to perform highly undetectable steganography, in: Proceedings of International Workshop on Information Hiding, 2010, pp. 161–177.

[32] V. Holub, J. Fridrich, Designing steganographic distortion using directional filters, in Proceedings of IEEE International Workshop on Information Forensics and Security, 2012, pp. 234–239.

[33] V. Holub, J. Fridrich, Digital image steganography using universal distortion, in: Proceedings of ACM workshop on Information Hiding and Multimedia Security, 2013, pp. 59–68.

[34] B. Li, M. Wang, J. Huang, et al., A new cost function for spatial image steganography, in: Proceedings of IEEE International Conference on Image Processing, 2014, pp. 4026–4210.

[35] J. Fridrich, J. Kodovský, Multivariate Gaussian model for designing additive distortion for steganography, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 2949–2953.

[36] V. Sedighi, J. Fridrich, R. Cogranne, Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model, in: Proceedings of SPIE, Media Watermarking, Security and Forensics, 2015, pp. 0H01–0H13.

[37] B. Li, M. Wang, X. Li, et al., A strategy of clustering modification directions in spatial image steganography, IEEE Trans. Inf. Forensics Secur. 10 (9) (2015) 1905–1917.

[38] T. Denmark, J. Fridrich, Improving steganographic security by synchronizing the selection channel, in: Proceedings of ACM Workshop on Information Hiding and Multimedia Security, 2015, pp. 5–14.

[39] Z. Wang, A.C. Bovik, H.R. Sheikh, et al., Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[40] G. Schaefer, M. Stich, UCID-An uncompressed colour image database, in: Proceedings of SPIE, Storage and Retrieval Methods and Applications for Multimedia, 2004, pp. 472–480.

[41] A. Olmos, F.A.A. Kingdom, A biologically inspired algorithm for the recovery of shading and reflectance images, Perception 33 (12) (2004) 1463–1473.

[42] http://imagedatabase.cs.washington.edu/, University of Washington, Object and Concept Recognition for Content-Based Image Retrieval.

[43] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (27) (2011) 1–27.

[44] B. Li, J. Huang, Y.Q. Shi, Textural features based universal steganalysis, in: Proceedings of SPIE on Security, Forensics, Steganography and Watermarking of Multimedia, 2008, pp. 681912.

[45] T. Pevný, P. Bas, J. Fridrich, Steganalysis by subtractive pixel adjacency matrix, IEEE Trans. Inf. Forensics Secur. 5 (2) (2010) 215–224.

[46] J. Fridrich, J. Kodovský, Rich models for steganalysis of digital images, IEEE Trans. Inf. Forensics Secur. 7 (3) (2012) 868–882.