Multimed Tools Appl (2016) 75:8319–8345
DOI 10.1007/s11042-015-2752-9

An adaptive multi bit-plane image steganography using block data-hiding

Tuan Duc Nguyen[1] · **Somjit Arch-int**[1] · **Ngamnij Arch-int**[1]

Received: 22 September 2014 / Revised: 21 April 2015 / Accepted: 15 June 2015 /
Published online: 2 July 2015

Abstract Embedding a secret message into the pixels of a cover image yields a visual distortion if these pixels belong to smooth regions. Thus, this prompted the development of some edge-based approaches in which only the edge pixels are used to hide secret bits. As a result, the visual quality of stego images is improved. However, the capacity is limited due to some unused regions in cover images. In this paper, an adaptive multi bit-planes image steganography using block data-hiding (MPBDH) is proposed. This method employs more than one bit-plane and applies an adaptive complexity threshold computation to select the complex regions of a cover image used in data hiding. Consequently, the embedding capacity and security performance are significantly improved in comparison with previous approaches based on pixel and block complexity. The results, which are obtained from experiments performed on 10,000 natural gray-images, indicate that the embedding capacity and security introduced in the proposed approach overcome the problems of previous approaches. The proposed approach is hence suitable for secure communications.

Keywords Muli bit-plane · Adaptive · Block data-hiding · Steganography

1 Introduction

In recent years, with the rapid development of network techniques, the amount of valuable data transferred via the Internet has increased. Steganography is employed in order to protect transferred data from security breaches. It aims to embed secret data into a digital cover object, such as a digital audio file, image, video, or even network traffic.

✉ Tuan Duc Nguyen
nguyenductuan1982@gmail.com

Somjit Arch-int
somjit@kku.ac.th

Ngamnij Arch-int
ngamnij@kku.ac.th

[1] Khon Kaen University, Khon Kaen, Thailand

Image steganography uses digital images as the media through which to convey secret data. The images used for hiding data are called cover images, and the images with data embedded are referred to as stego–images. After embedding the data, the pixels of the cover image are alternated; thus distortions are introduced.

In general, detectability (security), payload (embedding capacity), and image quality are the most important factors of image data hiding. A low detectability prevents the stego-images from being discovered by visual or statistical attack methods. A high payload indicates that more data bits are hidden, and better visual quality of an image means less distortion is introduced.

Least Significant Bit (LSB) replacement, in which the Least Significant Bits (LSBs) of each pixel in the host image are replaced by message bits, is the most popular steganography method. This method is easy to implement and it produces stego-images of an appreciable visual quality. Unfortunately, it is vulnerable to the steganalysis attack [6] due to the imbalanced number of even and odd pixels caused by the data hiding process. LSB matching (LSB-M) [12] reduces the detectability of LSB replacement by randomly adding or subtracting the pixel values by one according to the message bits. Nevertheless, LSB-M still causes the same amount of distortion as the LSB method [9].

To minimize the possibility of being determined by visual attack, Luo et al. presented edge adaptive image steganography based on LSB Matching Revisited [14] (EA_LSBMR). This scheme adaptively selects embedding areas according to the size of secret message and the difference between two consecutive pixels in a cover image. For lower payloads, only higher textured pixels are used, when the payload increases, more pixels in other regions are selected. However, this approach will employs all pixels (including pixels belong to smooth regions) in the cover image when the embedding rate is 1 bpp. As a result, at high embedding rates, the security against visual attack methods and visual quality of the stego-images produced by EA_LSBMR are lower than that at lower embedding rates.

In [22], Sabeti et al. introduced an approach in which the distortions are minimized by applying a two-phase process. At first, a cover image is formatted by setting the LSBs of all pixels to "0". This replacement reduces the number of embedding changes when a message bit is "0" as well. Then, the LSB-M is employed to hide secret bits in the complex pixels. Nevertheless, this approach is not secure against visual attack due to the pattern created by the formation process.

To increase the security against visual attack, a block-based steganographic algorithm called a Pixel Rearrangement based Steganography Algorithm (PRSA) was proposed [23], in which secret bits are embedded into a set of pixels by rearranging the location of the pixels in the considered set. In other words, secret bits are represented by a different pixel ordering in a set of pixels. Nevertheless, changing pixel's position causes more degradation in structural information of a cover image. To improve the security of hidden data against visual based attacks, in this approach, an adaptive block selection is applied to choose a textured area to hide a secret data. However, the embedding capacity is limited if the used cover image contains some large non-textured regions.

In [10], Jung et al. proposed a scheme in which image pixels are divided into edge and non-edge areas to improve the visual quality and capacity using an edge detection algorithm. For convenience, we use the term 'EDSI' when referring to this proposed approach in this paper. In EDSI, for edge regions, E_b secret bits are embedded into each pixel that belongs to the considered area. The value of E_b is predefined based on a threshold T. For non-edge areas, the number of secret bits embedded into a pixel is calculated by the difference between two

non-overlapping consecutive pixels in the block. As a result, the embedding capacity is increased. Nevertheless, hiding secret bits in non-edge areas increases the possibility of being detected by a visual attack method (such as LSB Enhancement) because the LSBs of the pixels in the flat regions have the same value ("1" or "0") [14]. Therefore, if we hide the secret message in these areas, the LSB of the stego-image would become more and more random, leading to visual differences between cover (contains smooth regions) and stego-images (presented as an embedding noise-like distribution). These noises are significantly enhanced by the LSB Enhancement attack method. Hence, the human vision system can distinguish the embedding noise in the resulting image generated by the LSB Enhancement process.

Additionally, the common weakness of the above approaches is the lack of ability to prevent extraction attacks. An attacker tries to export a hidden message from a suspected stego-image with all possible values of threshold T in three methods, EA_LSBMR, PRSA and CBL. For EDSI, the data extraction attack is simpler when the complexity thresholds are predefined.

In this paper, we introduce a multi bit-plane block data-hiding (MPBDH) approach, in which a secret message is embedded into noisy regions of cover image's bit-planes to increase the security of the hidden data. By employing a block data-hiding (BDH) algorithm introduced by Nguyen et al. [17] as an embedding method, the visual quality of the stego-images is guaranteed as well. The main objective of the proposed approach is to solve the existing drawbacks of the previous approaches by addressing the following problems:

- To minimize the possibility to be detected by visual attacks, in the proposed approach, secret bits are adaptively embedded into noisy areas of a cover image due to the less sensitive to changes in smooth regions of the human eye. The image's regions are selected based on the number of message bits to be embedded and the texture characteristic of a cover image. Furthermore, hiding a secret message in the textured regions leads to an increase in the perceptual quality of the stego-images.
- To overcome the existing limitation of embedding capacity in the previous methods, in the proposed approach, more bit-planes of the cover image are employed in the data hiding to increase the number of message bits that can be hidden in an image while still maintaining the visual quality of the stego-images. This advantage is obtained by applying an adaptive region selection for each used bit-plane separately. The adaptive region selection is performed based on the complexity threshold which is estimated in parameter estimation. Hence, at the low embedding rates, the high noise regions are used. For a higher payload, the complexity thresholds are adjusted in order to acquire more areas with lower texture characteristics to hide all the given secret bits.
- To reduce the distortions introduced by multiple bit-plane embedding method, block data hiding (BDH) method is employed to hide secret message bits to selected regions. It is because the BDH algorithm embeds secret bits to a block of cover bits by modification at most two bits. In addition, BDH algorithm presents several embedding change solutions, embedding noises are further minimized by choosing the solution that causes less distortion to cover images. Moreover, the high bit-planes are employed first and an adaptive adjustment is then applied to reduce more degradation caused by data hiding since more bit-planes are employed.
- To minimize the possibility of the hidden data being extracted by attackers, the default value of complexity thresholds, which allows user control the trade-off between the visual quality of stego-images and payload, can also be used as a security key to protect an

unseen message against extraction attacks. Without the knowledge of these default values, an attacker cannot determine how many blocks were selected at one bit-plane to hide secret message bits. The reason is that in parameter estimation phase, if an estimated complexity threshold of one bit-plane is lower or equal to the value of the predefined corresponding default threshold, the next bit-plane will be employed. To further enhance the security of hidden data against extraction attack, a *keystream*, which is produced by Advanced Encryption Standard (AES), is also used to transform a block of cover bits into another form before applying BDH to hide secret bits.

The remainder of this paper is organized as follows. In Section 2, the related works are presented. Then, in Section 3, the proposed method is introduced in detail. In Section 4, the experimental results are given, and finally, several conclusions are drawn in Section 5.

2 Related work

In this section, an approach based on block [8] and pixel complexity [7, 22] is discussed. The BDH [17] which is used as the embedding method in the proposed approach is also introduced.

2.1 A block complexity based data embedding

The principle of A Block Complexity based Data Embedding (ABCDE [8]) is the same as that of Bit-Plane Complexity Segmentation Steganography (BPCS) [11]. The data hiding process is performed by replacing the pixel data of noisy regions in an image with other noisy data obtained by converting data to be embedded. A complexity measurement is used in BPCS Steganography to identify the noisy regions in an image that are used to hide secret bits. It is a suitable measurement, but it is not always applicable. Therefore, two new complexity measurements, called run-length irregularity and border noisiness, are proposed and applied to ABCDE. If a block has large run-length irregularity and border noisiness at the same time, it is a complex block.

The run-length irregularity is defined based on the histogram of the run-length of both black and white pixels along a row or a column of a block. The border noisiness complexity measurement is computed based on the differences between adjacent binary pixel sequences in a block.

A cover image is divided into non-overlapping blocks of pixels. For each segmented block, the complexity is measured to identify which bit-plane is used to hide the secret bits.

Nevertheless, this approach cannot select the complex regions adaptively and automatically for each bit-plane of a cover image. This is because, in this method, certain control parameters need to be set manually, such as finding an appropriate section length for sectioning a stream of resource blocks and finding the threshold value for identifying complex blocks.

Moreover, the possibility of preventing an attacker from extracting the hidden data is low since an attacker can employ the predefined complexity threshold values introduced in [8] and travel through the suspected stego-image to extract the embedded bits.

2.2 A steganography method based on complex pixels

In this type of image steganography method [7, 22], the complexity of every pixel in an image is computed and then compared with a threshold to identify whether the considering pixel is complex or not.

In the CBL method [22], the complexity of pixels is measured based on the sum of absolute values of differences of the pixel with its 8-neighbors. The complexity is calculated by the following equation:

$$complexity(i,j) = \sum_{u=-1}^{1}\sum_{v=-1}^{1} |I(i,j) - I(i+u, j+v)|$$

where $I(i, j)$ is the pixel (i, j) in the cover image I, and u, v are the indexes to indicate the position of the neighboring pixels of $I(i, j)$.

Then the computed complexity is compared with an estimated threshold to identify which pixel is used in the embedding process. The threshold is estimated according to the length of the secret message and the number of pixels in the image. Assuming p is the percent of the pixels to be embedded, then the value of threshold T is selected such that at least p percent of the pixels are labeled as complex and are embedded with data. T is measured by the following equation:

$$T = \max\{t_0 \| \{(i,j) | complexity(i,j) \geq t_0\} | \geq (p*m*n/100)\} 0 \leq p \leq 100$$

The advantage of CBL is that the complexity of the pixels is not changed after the secret data is embedded into it by employing image formation. In this process, the LSBs of all image pixels are set to zero in order to have the same set of complexity values after the data hiding is performed. As a result, correction in the data extraction is guaranteed. However, image formation introduces some patterns in the resulting image that are produced by the LSB Enhancement attack [27]. In addition, employing the image formation leads to the increase of computational time for the data hiding. Therefore, this approach is not suitable to use in real-time systems.

Furthermore, the capacity provided is limited when a cover image contains some smooth regions. This is because the non-edge pixels are not used when the secret message is being embedded.

2.3 Block data hiding algorithm

In this section, binary block data-hiding, which hides a data byte d in a binary block $(m \times n)$, is introduced. The main idea of this algorithm is that the secret bits are carried by a binary block $(m \times n)$, in which at most 2 bits of the binary block are changed without requiring an additional matrix or vector.

Supposing that F is a binary block size of $m \times n$, and $d(0 \leq d \leq m \times n)$ is a data byte that is embedded into F, then S is denoted as:

$$S = \sum_{i=1}^{m}\sum_{j=1}^{n} F(i,j)((i-1)n + j)(\mod K) \quad (1)$$

where K is defined as $K = K^* + 1$ (with $K^* = m \times n$). To embed the value d into the binary block F (as shown in Fig. 1(a)), the process must be performed to determine the position of bits in F

that can be flipped to make the value of S equal to the value of d. In the extraction process, the value of d is calculated using Eq. 1 from the binary block taken from a stego object.

Example Let us consider the given binary block size of 5×3

From the given binary block F (Fig. 1(a)), S=2 (calculated using Eq. 1) and K=16. Assume that d=5, then b=2−5=−3, then flipping the bit at position (5, 1) produces $S'=S-b(\mathrm{mod}K)=-2(-3)(\mathrm{mod}16)=5$.

Lemma *For all integers* $d\in[0,K]$, *a maximum of two bits in block F need to be changed to transform F into F'.*

$$S' = \sum_{i,j}^{m}\sum_{i,j}^{n} F'(i,j)((i-1)n+1)(\mod K) = d(\mod K) \tag{2}$$

Suppose that Ω_1 is a set of the position of bits that satisfy:

$$\Omega_1 = \{(i,j) : F(i,j)=1, (i-1)n+j=b\} \cup \{(i,j) : F(i,j)=0, (i-1)n+j=K-b\} \tag{3}$$

Ω_2 is the set that contains the position of the pair of bits in which flipping the bits in this set makes the value of S equal to d. Ω_2 is defined as follows:

$$\begin{aligned}\Omega_2 \quad &= \{\{(i,j),(p,q)\} : F(i,j)=0, F(p,q)=1, i\times n+j=pn+q-b\} \\ &\cup\{\{(i,j),(p,q)\} : F(i,j)=1, F(p,q)=1, (i-1)n+j+(p-1)n+q=b(\mod K)\} \\ &\cup\{\{(i,j),(p,q)\} : F(i,j)=0, F(p,q)=0, (i-1)n+j+(p-1)n+q+b=0(\mod K)\}\end{aligned} \tag{4}$$

In Example, we can indicate $\Omega_1=\{(5,1)\}$ and

$$\Omega_2 = \{\{(1,1),(5,2)\},\{(4,1),(1,3)\},\{(1,1),(1,2)\}\{(2,1),(5,3)\}\}$$

Clause 1 *For any binary block F and given d, two sets* Ω_1*and* Ω_2*are not simultaneously empty.*

Clause 2 *In the binary block F, flipping the bit at* (i, j) *or the pair of bits at the set* $\{(i,j),(p,q)\}$ *causes* $S'=d(\mathrm{mod}K)$ *when* $(i,j)\in\Omega_1$*or* $\{(i,j),(p,q)\}\in\Omega_2$.

Detailed mathematical proof of Lemma, Clause 1, and Clause 2 was presented in [17].

An extract process calculates a hidden data byte d by the Eq. 1 for each cover block until a complete secret message is extracted.

3 Proposed approach

In this section, we propose a scheme that employs noisy regions in a bit-plane of an image to hide secret bits to improve the visual quality of stego-images and the security of the hidden data.

As shown in Fig. 2, the proposed approach is demonstrated as a secure channel for transferring sensitive information (such as a business contract, bank account numbers, etc.). In which a secret message is transformed into binary format and then encrypted by AES [5]. After that, these encrypted data are embedded into the selected digital image by MPBDH (which is described in Section 3.1).

0	0	1
0	0	0
1	1	0
1	1	0
1	1	0

(a)

0	0	1
0	0	0
1	1	0
1	1	0
0	1	0

(b)

Fig. 1 Binary block F (**a**) and F' (**b**) (after flipping)

After data embedding, the stego-image is then transferred to the receiver via the network environment. The sender can upload it to Online Public Image Sharing, which keeps the stego-image as the original version. In another way, the sender can embed the stego-images into an email and then send it to the receiver.

To guarantee that only the authorized receiver can extract the hidden message, the generated key, which is used in the data hiding process, is also employed to transform the cover bits (from the cover pixels) to another form before embedding secret bits into them. Therefore, an attacker cannot recover the unseen message from stego-images without the knowledge of encrypted secret key (which is used to regenerate the used *keystream*).

The flow diagram of the proposed approach is illustrated in Fig. 3, in which the image regions are adaptively selected to use in the embedding process. The region selection is executed based on the two parameters, complexity threshold array and the number of bit-

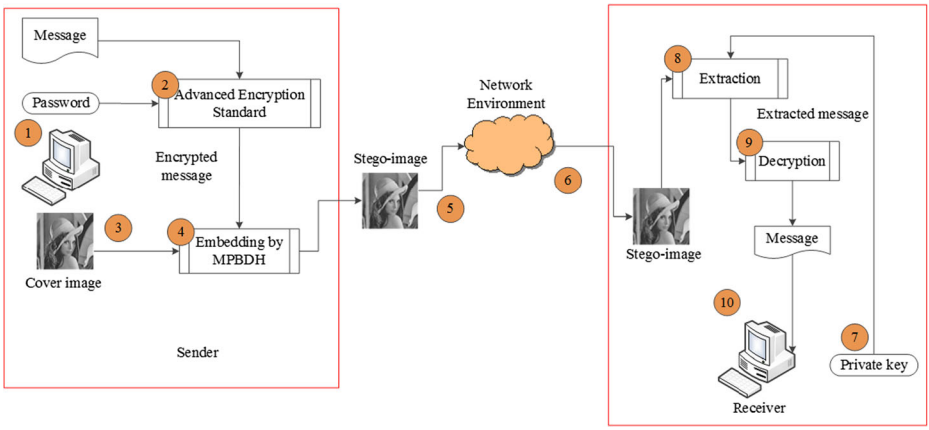

Fig. 2 Conceptual Framework of the proposed approach

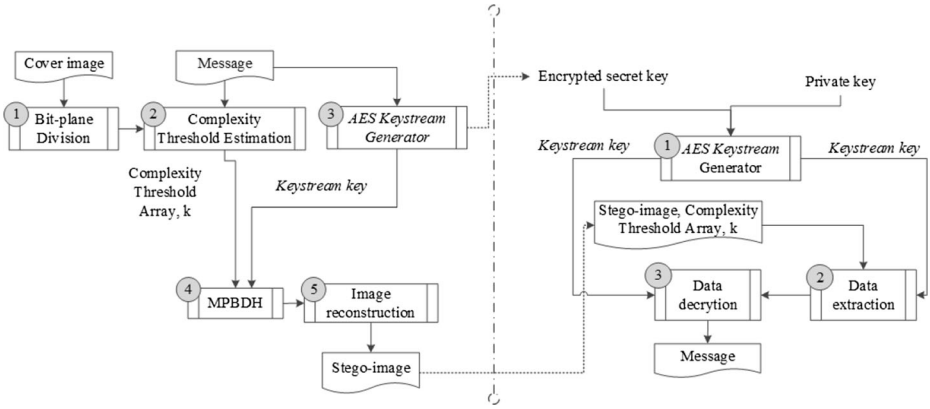

Fig. 3 A flowchart of the proposed approach, including the data embedding and data extraction process

planes k, which are estimated by Parameter Estimation (the process is presented in Section 3.5). In each iteration of parameter estimation, if the number of texture regions that are selected with the current values of the estimated parameters is enough to embed the given message, the data embedding is performed. Otherwise, the parameter estimation process needs to adjust the value of parameters until the selected regions are available to carry all bits of the given message.

For the high embedding rates, more bit-planes are used to hide the given secret data, the secret bits are spread out into different bit-planes. Therefore, the number of secret bits embedded into the different bit - planes is not equal and this number is identified by the default complexity threshold for each bit-plane. In the proposed parameter estimation, these default thresholds were slightly adjusted in comparison with their original state in ABCDE [8] in order to hide more secret bits to lower bit-planes. This means the value of these parameters are different for different cover image and secret message.

To guarantee the correction of extraction process, these estimated parameters are first encrypted and then embedded into unaltered regions of the stego-image or transferred to the receiver separately.

3.1 Multi bit-planes block data-hiding

A cover image is divided into blocks (8×8) pixels). Then the complexity values of k bit-planes are measured to identify which bit-plane of a block is used to hide the secret bits, where k is the number of bit-planes employed to embed the given message bits.

The selected block at the bit-plane k^{th} is transformed to 1D pixels sequence (sized 1×64) before being divided into nine segments sized 1×7. The transformation is performed by employing Hilbert filling curve [21]. Then, these divisions (blocks of bits) are XORed with the key bits (from *keystream key*) before applying the BDH to hide a set of bits from a message to a segment. Each set of bits contains three bits because the BDH employs seven cover bits to hide three message bits by changing two bits in the segment at most.

The embedding process is performed from the bit-plane k^{th} down to 1 in order to employ an adaptive adjustment to minimize the distortion of the cover images. Moreover, please note that

since the BDH offers more than one modification solution that can be processed to hide secret bits, the embedding solution, which introduces fewer changes to the cover pixels, is employed. The embedding noise caused by the data hiding process is hence further reduced.

Algorithm 1: Multi bit-planes block data-hiding

Data: Gray-scale image I, secret message M, Complexity Threshold Array TH, *keystream key* and k

Result: stego-image I'

```
1  M ← Encryption(M, key);
2  while (curX < imgW && curY < imgH) do
3      B ← getBlockOfPixels(I, m, n, curX, curY);
4      for i ← k down to 1 do
5          kB ← getNextBlockKeyBits(key);
6          F ← B(i);                          // get bits at the bit-plane i^th of B
7          COMP ← compEstimation(F);
8          F ← F ⊕ kB;
9          if (COMP ≥ TH(i)) then
10             BF ← blockSegmentation(F);
11             grpArr ← getGroupOfBitsArray(M, 7);
12             for j ← 1 to 7 do
13                 grp ← grpArr(j);
14                 FP ← blockDataHiding(BF(j), grp, kB);
15                 BF(j) ← adaptiveAdj(FP);
16         end
17         F ← reconstructBlock(BF);
18         F(64) ← 1;
19     else
20         F(64) ← 0;
21     end
22 end
23 B(i) ← F;
24 saveArrPix2Img(B, I');
25 if end of message stream then
26     exit;
27 end
28 end
```

The last bit of each selected binary block of the considered bit-plane is used as an indicator to guarantee that the extraction process is correct when the complexity of the considered block is lower than the threshold after embedding.

3.2 Adaptive region selection

Embedding secret bits into texture areas introduces less noticeable degradation to stego-images. It is because the effect caused by data embedding is illustrated as the noise. Therefore, in the proposed approach, the embedding regions are selected from high texture characteristic areas of the cover image.

At first, the cover image is divided into a non-overlapping block sized 8×8, then the complexity of the bit-planes of the block are measured. The number of considered bit-planes is based on the value of the parameter k (which is estimated by parameter estimation).

If the complexity of the bit-plane i^{th} of the considered block is larger or equal to the measured complexity threshold i^{th} in the estimated complexity threshold array, this block is used to hide the secret bits. Otherwise, the block of bits at lower bit-planes are considered. Please note that although the parameter estimation is started with the LSB bit-plane, in data hiding the k^{th} bit-plane (k is the number of bit-planes will be used in data hiding) is employed first.

As can be seen from Fig. 4, there are more blocks (marked by red squares), which are selected when the payload is increased. At a high embedding rate, the complexity threshold is reduced to obtain more regions to use in data hiding. For an embedding rate of 0.4 bpp, the number of selected blocks is the same as the embedding rate of 0.3 bpp. It is because at this bit-plane, there are no more

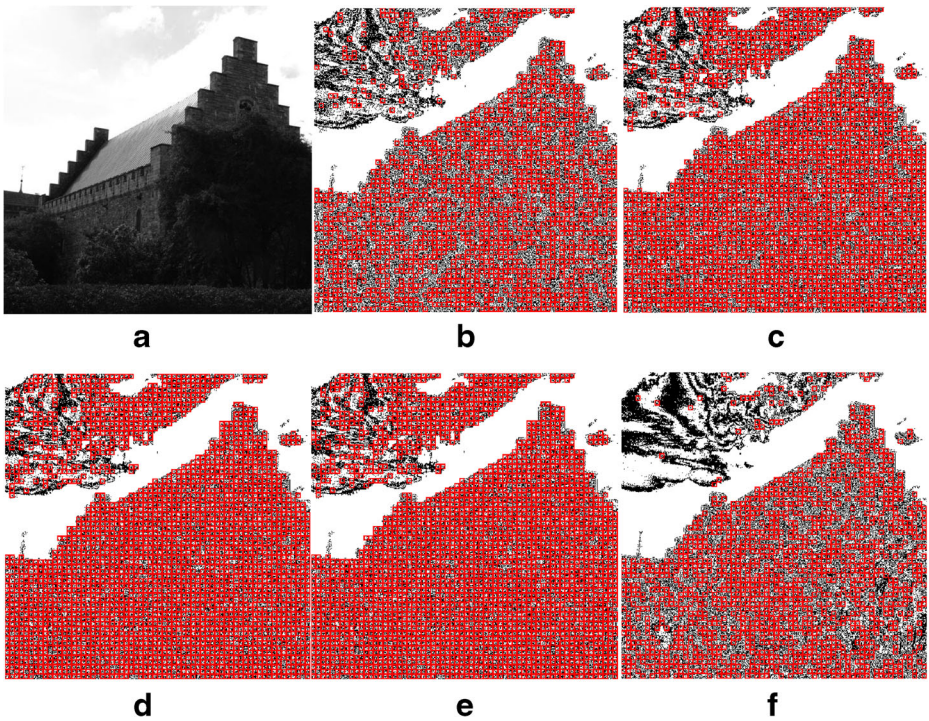

Fig. 4 Selected block locations for different payloads of the least significant bit-plane. **a** Cover image, Payload at 0.1 bpp (**b**), 0.2 bpp (**c**), 0.3 bpp (**d**), 0.4 bpp (**e**) and selected blocks at the 2nd bit-plane for embedding rate of 0.4 bpp (**f**)

blocks of pixels that can be selected with the estimated complexity threshold for this bit-plane. Therefore, the higher bit-plane is considered and the selected regions of the 2nd bit-plane are illustrated in the Fig. 4 (f). Meanwhile, employing multi bit-plane is not available in the adaptive block selection which is presented in PRSA [23].

From Fig. 4, it can be observed that the smooth regions (the sky between cloud and house, and some other flat areas) in different bit-planes of a cover image are not employed to carry the secret message bits even at high embedding rates.

3.3 Data embedding

As shown in Fig. 3, the data embedding process consists of the following five steps:

- Step 1. A cover image *I* is partitioned into bit-planes. A given secret message *M* is embedded into each bit-plane separately.
- Step 2. The parameter estimation, which is presented in Section 3.5, is performed to measure the complexity threshold for each bit-plane. This bias is measured adaptively according to the length of the given message. If the size of the secret message is small, blocks with high complexity are used. Otherwise, blocks with lower texture characteristics are employed.
- Step 3. AES is implemented in Counter (CTR) mode. In this chaining mode, AES works as a stream cipher to encrypt a give message with non-fixed length (it doesn't have to be a multiple of the cipher block size). A secret key, which is used as input key to AES to produce *keystream* key, is random generated with the length of 32 (this value of length indicates that data and cryptographic key in blocks of 256-bits). Then this secret key is encrypted by public key cipher RSA [20] before sending it to receiver. This process reduces a possibility that the secret key can be decrypted by an attacker to regenerate the used *keystream*. The encryption is performed by XORing the generated *keystream key* with binary sequence of the given message.
- Step 4. In this step, the MPBDH is employed to hide the encrypted message in the bit-planes of the cover image.
- Step 5. In this phase, the bit-planes with the secret message embedded within them are re-constructed to create a stego-image before being sent to the receiver via the network environment.

3.4 Data extraction

At the receiver's side, the data extraction process constitutes the following three steps:

- Step 1. A private key provided by the receiver is used to decrypt the encrypted secret key, which is used as an initial key in *keystream* generation of AES. The generated *keystream key* is then employed in the extraction stage.
- Step 2. The parameters (complexity threshold array and number of used bit-planes) are extracted from the stego-image or obtained from the sender via predefined communication. Then the data extraction travels through the stego-image to find a block which its complexity is greater or equal to the complexity threshold and the 64^{th} bit is "1". The bitwise XOR operator is applied to the selected block of current bit-plane and key data bits (from *keystream key*). The result (block of bits) is used as input to block data-hiding algorithm to extract a group of three hidden bits. There are nine groups extracted from the selected block (8×8 binary pixels).

– Step 3. The extracted data is then XORed with the *keystream key* generated by AES to transform the encrypted message into the secret data that the sender embedded into the cover image.

3.5 Parameter estimation

To reduce the visible distortions in a stego-image, a secret message should be embedded into noisy regions, since the human eye is not sensitive to slight alterations in these regions. A complexity measurement is introduced to determine whether a block is complex or not. There are two new complexity measurements proposed in ABCDE [8] in order to evaluate the texture characteristics for a bit-plane of a cover image. They are run-length irregularity and border noisiness. In the proposed approach, only the run-length irregularity (denoted as *beta*) measure is used to reduce the computational time. In ABCDE, the threshold values of the complexity measure are independently fixed for each bit-plane. As a result, this approach cannot adaptively select the noisy regions according to the length of the given message.

In the proposed approach, the complexity threshold is adjusted adaptively to select the noisy regions with texture characteristics as high as possible. The adaptive complexity threshold estimation is presented in Algorithm 2. In this algorithm, the threshold is calculated separately for each bit-plane and the number of used bit-planes are based on the length of the input secret message. At first, the value 1.0, which is the maximum value of run-length irregularity, is set to *beta*. The number of regions at the considered bit-plane, which has an estimated complexity measure is greater or equal to *beta*, is counted. If the number of available regions is smaller than the number of regions need to be selected, the parameter is adjusted by a value of a variable *step* until the selected texture regions is enough to carry the given secret message. The variable *step* is employed to control the tradeoff between speed and capacity. If the value of the *step* is small, more blocks are considered for evaluation as an embedding unit. Nevertheless, the computational time is increased.

The default beta threshold values (that are slightly adjusted in comparison with them in ABCDE) are employed to guarantee that the selected regions are as high as possible in terms of texture characteristics. These default threshold values are estimated in such a way that all the bits of a secret message can be hidden in most cover images with more secret bits embedded into low bit-planes.

The estimation process is finished when the estimated threshold values (which are stored in *beta_esti*) are available to select a sufficient amount of textured regions to hide the given secret message. In each iteration, the measured complexity (*beta_comp*) of a binary block in the considered bit-plane is compared with the threshold *beta* to count the number of texture blocks selected with this threshold. If the number of selected blocks is larger than or equal to the number of needed blocks, the value of *beta* is then stored in output complexity thresholds *beta_esti* for the current bit-plane. Then the estimation phase continues with the next bit-plane if there are still more secret bits that need to be embedded or the number of used bit-planes is smaller than eight. Otherwise, if the value of *beta* is still greater than the default value *beta_def*(k) of this variable for the current bit-plane k^{th}, *beta* is then reduced an amount of *step* and the counting is started again with the new value of *beta*. In the case of the value of *beta* being below *beta_def*(k), *beta* is set to *beta_esti*(k), and the next bit-plane is processed if there are more secret bits to be embedded. Finally, the estimated complexity thresholds are stored in an array *TH*, and then this array is used to select the regions in the cover image to carry the message bits.

This process supports to adaptively select the texture regions to carry the secret message bits, while in previous approach ABCDE, the default complexity thresholds

are employed. As a result, lower texture characteristic regions are used to embed a secret message even at a low embedding rate. In addition, in ABCDE, for cover images that contain some smooth areas and with one of them being large, the selected textured regions are not enough to hide the given message due to the use of fixed complexity thresholds.

Algorithm 2: Parameter Estimation;

Data: Gray-scale image I, length of message L

Result: Estimated Complexity Threshod array TH, number of bit-planes will be used k

```
TH ← []
k ← 1;
beta ← 1.0;
step ← 0.05;
beta_def ← [];
beta_esti ← [];
total ← [];
while (true) do
    if end of image then
        break;
    end
    B ← getBlockOfPixels(I, m, n, curX, curY);
    F ← getBitPlane(B, k);
    beta_comp ← compEstimation(F);
    if (beta_comp ≥ beta) then
        total(k) ← total(k) + 1;
    end
    if (sum(total) ≥ numOfBlockNeeded) then
        beta_esti(k) ← beta;
        break;
    else
        if (beta > beta_def(k)) then
            beta ← beta − step;
            total(k) ← 0;
        else
            if (total(k) > 0) then
                beta_esti(k) ← beta;
            end
            if (k < 8) then
                beta_esti(k) ← beta;
                k ← k + 1;
                beta ← 1;
            else
                break;
            end
        end
    end
end
TH ← beta_esti;
```

Furthermore, in the proposed parameter estimation, by adjusting the default value of complexity thresholds, we can control the tradeoff between the visual quality of the obtained stego-images or the embedding capacity. To select more blocks of pixels to hide the secret message, these default thresholds are reduced and the default values are increased to improve visual quality of the stego-images.

4 Experimental results and discussion

In this section, some experiments that measure the visual quality, embedding capacity, security, and embedding efficiency of the proposed approach were performed. Then the results were presented to demonstrate the effectiveness of our proposed algorithm compared with previous algorithms. There are two image databases used to examine the proposed approach's performance: the image database Break our Steganography System [24] (BOSS) and Signal and Image Processing Institute (SIPI) [1]. BOSS contains 10,000 gray-scale images with a size of 512×512 pixels in PGM format. These images were converted to bitmap (BMP) format using the command *imwrite* of Matlab. Although SIPI image database consists of color and gray images, only the gray-scale images were used in the experiments.

The secret message bits were generated by a pseudorandom number generator to ensure that the probabilities of bit "1" and "0" in the message are identical.

In the following experiments, for comparison, the BOSS image database (which contains 10,000 natural images) is used to hide a same amount message by the proposed methods, Tri-way Pixel-Value Differencing (TPVD)[4], EA_LSBMR, the pixel complexity based method CBL, and two block complexity based data hiding methods EDSI and PRSA.

4.1 Embedding capacity and image visual quality discussion

In general, in the steganography approaches based on edge detection or texture characteristic of image regions, a noisy block of pixels is selected to hide a secret message. Therefore, the embedding capacity depends on the texture feature of a cover image.

CBL method uses the sum of absolute values of differences between the considered pixel and its neighbors as the value of the pixel's complexity. According to the pixel's neighborhood defined in Fig. 2 (Section 3.3 of [22]), the complexity of pixels, which lay on the first and last row, the first and last column in a cover image, is equal to zero. The reason is that it has no neighbor in the previous position for the pixels in the first row and the next position of the pixels in the last row. Hence, these pixels are unusable in the embedding process. As a result, the available payload is reduced.

Moreover, with an image containing some smooth regions, one or some of which is large, the number of pixels whose complexity is equal to zero is very high. Thus, the embedding capacity provided by CBL is further reduced.

PRSA chooses the areas with a high frequency value to hide the secret bits because these regions present high texture characteristics. Therefore, the 2D Discrete Cosine Transform (DCT) is applied to blocks of 9×9 pixels to identify whether the considered block is a smooth or textured region. In data hiding, each selected block is segmented into non-overlapping sets of n pixels (b_k). According to the principle introduced in [23], if all the pixels in a set (b_k) are different to each other and $\max(b_k) - \min(b_k) < TH$ (threshold), the set is used for embedding. As a result, some sets of pixels are not used to hide the secret bits in cases where the block has very high texture characteristic. Hence, the payload of this approach cannot reach 1 bpp.

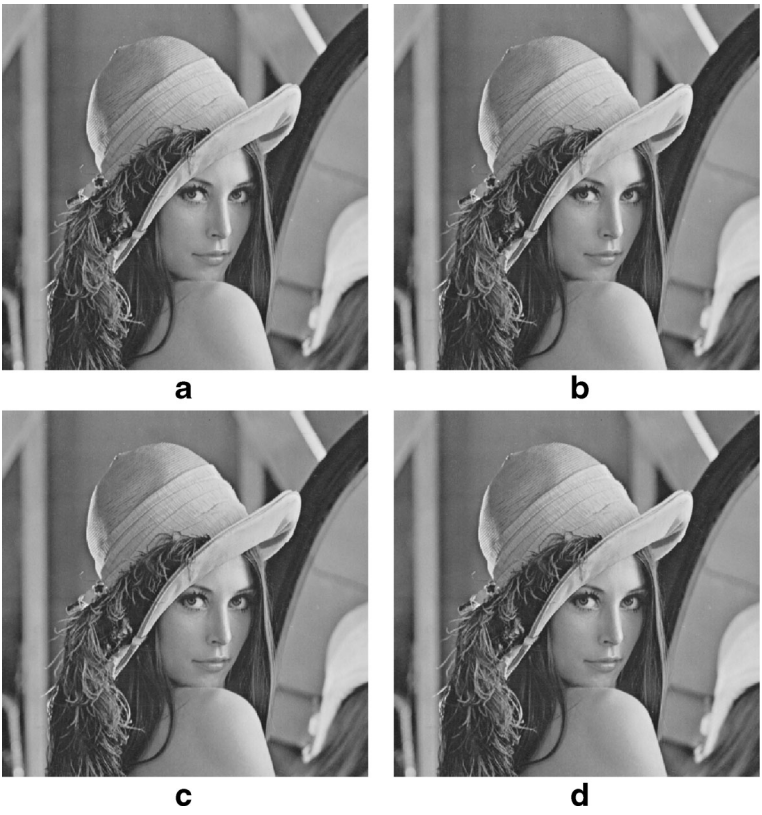

Fig. 5 **a** Cover image, **b**–**d** stego-images at embedding rate of 0.10 bpp, 1.00 bpp, and 1.5 bpp

Different to the two above methods, EA_LSBMR selects textured image's regions to adapt to the length of the given message. Therefore, this method can achieve an embedding rate of 1 bpp by employing all pixels in a cover image to hide a secret message. Nevertheless, the security against being discovered by visual attack of the stego-images produced by this approach is reduced.

Table 1 shows the capacity and PSNR of the proposed method compared with that of the previous method EDSI. In this experiment, parameter T is set to 4 in order to employ more

Table 1 Capacity and PSNR comparison between EDSI and MPBDH with images from USC-SIPI image database

Cover image	EDSI (T=4,Eb=2)		MPBDH	
	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)
Lena	389,976	44.88	393,288	46.10
Baboon	394,790	44.19	393,298	46.34
Tiffany	388,019	45.09	393,326	45.94
Peppers	391,128	44.84	393,629	46.61
Man	390,718	44.49	393,729	45.17
Boat	392,058	44.76	393,228	46.38

blocks of pixels in the data hiding process of EDSI and two secret bits are embedded into a selected block. Region selection with the parameter $T = 4$ is applied to all used cover images leads to the reducing of the visual quality and security of the stego-images. It is because, for a cover image with a low texture characteristic, modifying low-noise image regions (the gradient magnitude G is 4) may result in visible degradation in stego-images. Furthermore, for some cover images, the number of texture areas, which has G is larger or equal to 4, is small, the embedding capacity is limited.

For MPBDH, the complexity threshold for each bit-plane and the number of used bit-planes are adaptively measured according to the length of a given message. This means for different cover images, the complexity thresholds and the number of used bit-planes are not same. This is because each cover image has its own different texture characteristic. As a result, the proposed approach MPBDH can hide more secret bits in the same cover image when compared with EDSI.

As we can see in Table 1, the stego-images, which are produced by MPBDH, have a higher PSNR value than that of the stego images created by EDSI although there are more secret bits embedded into stego-images by MPBDH. This means that the quality of stego-images introduced by the proposed method is superior to that of the previous method EDSI.

Figure 5 presents a standard cover image (*Lena* image in SIPI) and the corresponding stego-images that were created by the proposed approach method MPBDH at different embedding rates. It is observed that there are no significant visual distortions caused by the proposed approach even at a high data rate (1.5 bpp).

In general, the degradations caused by data hiding are difficult to be seen by the human eye when this unseen data is embedded into noise regions (high texture characteristic regions). Therefore, to evaluate a performance of the proposed scheme in perceptual comparison with previous approaches, three visual quality metrics such as Peak Signal-to-Noise Ratio (PSNR), Weighted PSNR (wPSNR) [25], and Structure SIMilarity index (SSIM) [26]) are measured from 10,000 cover images and their corresponding stego-images. The PSNR is a statistical measure commonly used in image steganography for comparing the quality between the cover image and stego-image. Stego-images with higher PSNR are evaluated as better in terms of visual quality. wPSNR is an image quality metric which gives importance to the similarity of edges in the images. Hence it is widely employed to measure the visual quality of stego-images generated by edge based steganography [2, 19]. wPSNR is calculated by using the following equation:

Table 2 Average PSNR values of 10,000 stego-images produced by the proposed method (MPBDH) and previous approaches at embedding rates in the range of [0.10–0.40] bpp

Payload (bpp)	PSNR (in dB)						
	TPVD	EDSI	PRSA	CBL	EA_LSBMR	MPBDH	Improvement
0.10	45.89	57.08	53.15	58.08	61.54	62.65	1.80 %
0.15	43.91	55.32	48.08	56.32	59.59	60.90	2.20 %
0.20	42.45	54.07	44.20	55.04	58.30	59.64	2.30 %
0.25	41.20	53.11	41.08	54.00	57.17	58.68	2.64 %
0.30	40.20	52.31	38.18	53.30	56.39	57.88	2.64 %
0.35	39.36	51.65	35.54	52.61	55.56	57.22	2.99 %
0.40	38.43	51.07	33.42	51.98	54.92	56.64	3.13 %

Table 3 Average wPSNR values of 10,000 stego-images produced by the proposed method (MPBDH) and previous approaches at embedding rates in the range of [0.10–0.40] bpp

Payload (bpp)	wPSNR (in dB)						
	TPVD	EDSI	PRSA	CBL	EA_LSBMR	MPBDH	Improvement
0.10	63.41	69.79	71.80	73.29	66.30	78.05	6.49 %
0.15	61.40	68.05	66.33	71.48	65.70	76.27	6.70 %
0.20	59.89	66.82	63.49	70.20	64.77	75.01	6.85 %
0.25	58.60	65.89	61.06	69.15	64.44	73.99	7.00 %
0.30	57.58	65.11	58.47	68.44	63.81	73.21	6.97 %
0.35	56.71	64.48	55.77	67.77	63.54	72.53	7.02 %
0.40	55.75	63.95	52.46	67.14	62.92	71.96	7.18 %

$$wPSNR = 10\log_{10}\frac{\max(x)^2}{\|NVF(x'-x)\|^2}$$

where x is the cover image, and x' is the stego-image. NVF is the noisy visibility function [25] of the image.

According to the limitation of the embedding capacity of the PRSA method, the values of three metrics (PSNR, wPSNR, and SSIM) are divided into two categories. The first one consists of six methods (TPVD, CBL, EDSI, PRSA, EA_LSBMR and the proposed approach MPBDH) for the embedding rate in the range of [0.10–0.40] bpp. This is because that PRSA cannot embed a payload that is larger than 0.40 bpp for many images in BOSS. The second category presents the values of three metrics of the three approaches (TPVD, EDSI, and MPBDH) at embedding rates from 1.00 bpp to 1.50 bpp because CBL and EA_LSBMR cannot achieve an embedding rate of 1 bpp or higher. In this experiment, for the EDSI method, two scaling factors are set to $X = 1$ and $Y = 1$ in order to produce cover images with the same dimensions as the cover images used in other methods.

Table 2 presents the average PSNR values obtained from 10,000 stego-images generated by MPBDH and previous approaches for payloads of ranges [0.10–0.40] bpp. As can be seen, the MPBDH offers better image quality than the four previous approaches under various payloads. From this table, we can see that at the embedding rate of 0.40 bpp, the PSNR value of the proposed approach is still higher than that of PRSA when the payload of this method is 0.10 bpp.

The last column in Table 2 shows the improvement percentage of MPBDH in comparison with EA_LSBMR (the method has the highest image quality compared with the remaining approaches). The steady increase in these percentages indicates that the visual quality of the stego-images introduced by the proposed approach is better than that which was introduced by previous approaches. This result is achieved by employing the BDH that introduces a high embedding efficiency (the average number of secret bits per one embedding change) to hide secret data in complex regions.

The average wPSNR values measured from the cover images (in BOSS image database) and their corresponding stego-images (introduced by the proposed approach MPBDH and previous methods at embedding rates in the range of [0.10–0.40] bpp) are presented in Table 3. The last column of Table 3 presents the percentage of improvement of MPBDH in comparison with CBL.

Table 4 Average SSIM values measured from 10,000 cover images and their corresponding stego-images which are produced by the proposed method and previous approaches under payloads ([0.10–0.40] bpp)

Payload (bpp)	SSIM						
	TPVD	EDSI	PRSA	CBL	EA_LSBMR	MPBDH	Improvement
0.10	0.9952	0.9990	0.9965	0.9998	0.9998	0.9998	0.00 %
0.15	0.9919	0.9984	0.9915	0.9997	0.9996	0.9997	0.00 %
0.20	0.9887	0.9977	0.9847	0.9996	0.9994	0.9996	0.00 %
0.25	0.9853	0.9971	0.9758	0.9995	0.9991	0.9995	0.00 %
0.30	0.9818	0.9966	0.9629	0.9993	0.9989	0.9993	0.00 %
0.35	0.9782	0.9960	0.9448	0.9991	0.9986	0.9992	0.01 %
0.40	0.9745	0.9954	0.9198	0.9989	0.9979	0.9991	0.02 %

In contrast with EDSI, PRSA employs an adaptive block selection for different cover images, therefore, the stego-images, which introduced by PRSA, have higher wPSNR values than those of EDSI for different payloads. For EA_LSBMR, the wPSNR values are only higher than that of TPVD for different embedding rates. This is because EA_LSBMR identify embedding units by measuring the difference between two pixels in a pair of pixels. Therefore, in case of a pixel need to be modified is belong to a flat region, the stego-image will introduce a low wPSNR value. In addition, if a cover image contains several flat regions and one of them being large, all pixels of the cover image are used as embedding unit in data hiding. As a result, the wPSNR value of stego-images, which were created by EA_LSBMR method, is further reduced. Please note that the wPSNR value at embedding rate of 0.4 bpp is included the stego-images with unseen message hidden by this approach without applying adaptive selection.

In the proposed approach MPBDH, there are more high complex blocks chosen to embed the secret message due to the use of multi bit-planes and the application of the adaptive complexity threshold measurement for each bit-plane separately. As a result, high wPSNR values are obtained for the proposed approach under various payloads in the range of [0.10–0.40] bpp.

In the next experiment, SSIM, which is a metric for measuring the similarity between two images, is employed to evaluate the change in structural information of stego-images caused by the embedding process. An SSIM value close to 1.0 indicates that the structure of a stego-image has a similar resemblance to the original one. The obtained SSIM values of the stego-images introduced by the proposed method and previous approaches are shown in Table 4. The last column of the table presents the percentage of improvement of MPBDH over CBL (the method with the highest SSIM values in comparison with the remaining methods excluding the proposed method).

As we can see in Table 4, for the embedding rates in the range of [0.10–0.30] bpp, the SSIM values of the stego-images introduced by the proposed approach and CBL are the same. For the higher payloads (0.35–0.40 bpp), the obtained SSIM values of the proposed method are slightly higher than those of CBL. The reason given for this is because, for higher embedding rates employed, more secret bits are embedded into the stego-images, so more embedding noises are introduced. Therefore, for high embedding rates, the stego-image generated by the CBL method cannot maintain the same structural information as the original cover image. In contrast with CBL, in MPBDH, the adaptive block selection is applied to choose the areas with

a high texture characteristic to carry the secret bits. Therefore, the change in structural information of the stego-image is minimized. In addition, employing the BDH method in the proposed approach leads to a reduction of distortions when the BDH can hide the same amount of secret bits by performing a smaller number of embedding changes in comparison with the LSM-M method used in CBL. As a result, the structural information of the stego-images produced by MPBDH is less changed than that of CBL at the high embedding rate.

In EA_LSBMR approach, an adjustment is applied to modified pixels due to the value of some pixels is greater than 255 or lower than 0 (for example a value 256 is set to 252 or −1 is become 3) after embedding. Additionally, the stego-pixels are also re-adjusted if the difference between it and corresponding pixel is lower than threshold *T*. Consequently, more degradations in structural characteristic of stego-images are introduced. Hence, the SSIM values of stego-images that were created by EA_LSBMR are lower than that of the proposed method MPBDH.

For PRSA, although this approach also employs the adaptive region selection to identify embedding units (block of pixels) to hide a secret message, the change in structural information is high. The reason is that this method hides secret bits by swapping pixels in the selected block. Hence, more secret message bits are hidden, more pixels are swapped and the SSIM values are decreased rapidly even at lower embedding rates.

It can be observed from Table 5, for the high embedding rates in the range of [1.00–1.50] bpp, although the PSNR values are decreased, the PSNR values of the proposed approach are still superior to other methods for different payloads. The percentages of improvement, compared with the EDSI method, are presented in the last column of this table. As we can see, in contrast with the improvement percentages shown in Table 3, these values for the high embedding rates are lower than those at the lower embedding rates. This is because larger degradations are introduced at the high embedding rates. Therefore, the proposed approach cannot maintain the high image visual quality as with the lower payloads.

4.2 Security

4.2.1 Visual attack

In this section, the security performance of our proposed scheme against visual attack is discussed. In general, data embedding causes changes in the LSBs of the pixels in a cover

Table 5 Average PSNR and wPSNR values of 10,000 stego-images produced by the proposed method (MPBDH) and previous approaches for different embedding rates (from 1.0 bpp to 1.5 bpp)

Payload (bpp)	TPVD		EDSI		MPBDH		Improvement	
	PSNR	wPSNR	PSNR	wPSNR	PSNR	wPSNR	PSNR	wPSNR
1.00	29.19	46.59	47.12	60.53	51.48	66.70	9.26 %	10.19 %
1.10	28.68	46.12	46.71	60.19	50.86	66.07	8.89 %	9.77 %
1.20	28.16	45.60	46.34	59.88	50.33	65.49	8.61 %	9.36 %
1.30	27.68	45.14	45.99	59.60	49.62	64.73	7.88 %	8.61 %
1.40	27.37	44.85	45.68	59.33	48.77	63.90	6.76 %	7.71 %
1.50	27.10	44.59	45.38	59.08	48.08	63.20	5.94 %	6.97 %

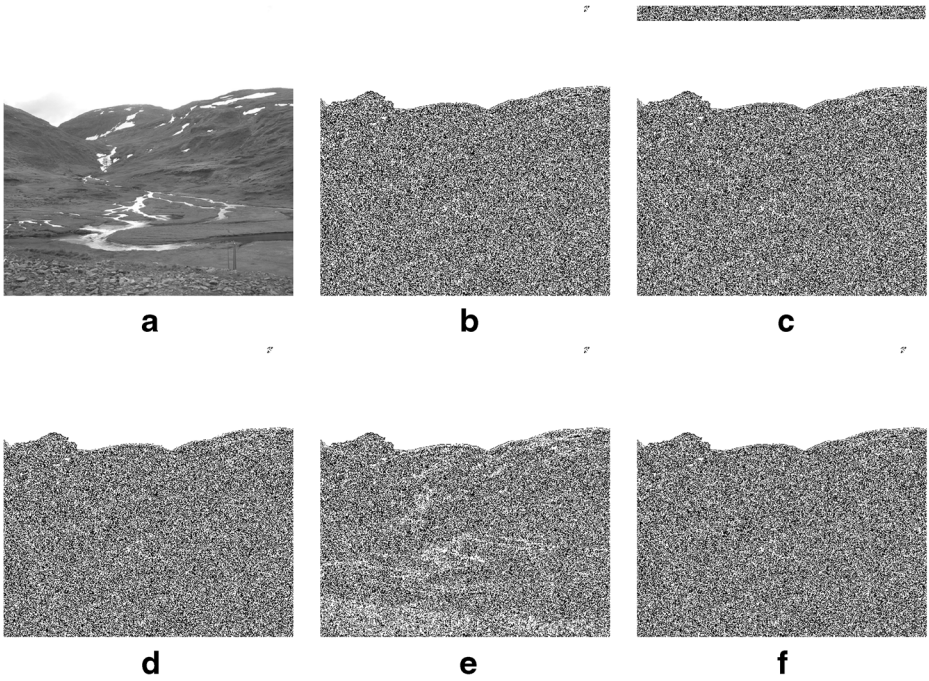

Fig. 6 **a** Cover image and its LSB (**b**), **c**–**f** LSB of stego-images using the four steganographic approaches and proposed approach at embedding rate of 0.10 bpp; **c** EDSI, **d** PRSA, **e** CBL and **f** MPBDH

image. Therefore, the idea of visual attacks is to remove all seven high-level bits for each pixel except the LSB bit-plane. These LSBs are then enhanced by setting a value "1" to 255 while keeping a value "0" as the original. This process makes the patterns, which are introduced by data hiding, become visible. As a result, these patterns in the resulting image can be recognized easily by the human eye.

LSB Enhancement is a visual attack method performed by a StegSecret tool [16] to obtain the enhanced LSB planes of all pixels in the stego-image created by MPBDH and previous approaches at the embedding rate of 0.10 bpp. The stego-image and its enhanced LSB bit-planes are illustrated in Fig. 6.

As can be seen in Fig. 6(c) (the enhanced LSB bit-planes of stego-images introduced by EDSI), the smooth region (sky) is contaminated to hide a message while this region in obtained LSB bit-planes of other stego-images is kept as in the original cover image. Nevertheless, the CBL method causes some changes to the LSBs of all pixels in the cover image. This process introduces some visible patterns on the resulting image (Fig. 6(e)).

In the next experiment, we hide a message with a payload of 1.50 bpp in cover images (from BOSS) to demonstrate the possibility of defending against the visual attack of the proposed approach at the high embedding rates. In this experiment, only the EDSI method is employed in comparison with the proposed approach because it can reach a payload of 1.50 bpp for all images in BOSS. The resulting images from the LSB Enhancement attack are illustrated in Fig. 7.

As can be seen in Fig. 7, the LSB planes of the stego-images generated by the proposed approach are completely the same as those of the original cover images. From Fig. 7 (d) and

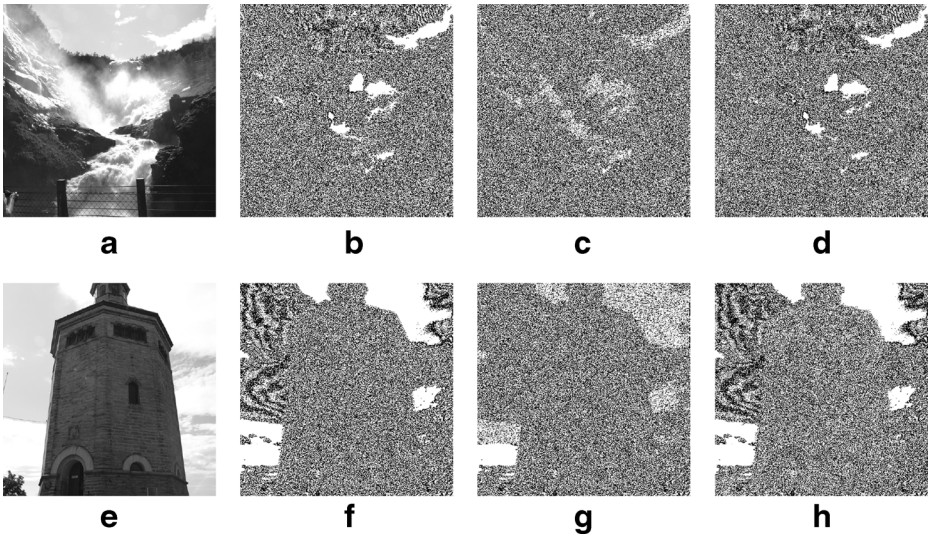

Fig. 7 The cover images (**a**, **e**) and their LSB planes (**b**, **f**), (**c**, **g**) LSB of the stego-images of EDSI and (**d**, **h**) LSB of the stego-images of the proposed method with the same embedding rate of 1.50 bpp

(h), it can be observed that the proposed method has the ability to avoid hiding a secret message in the flat regions such as the sky and shadowy parts. While in the LSB planes of the stego-images generated by EDSI, the smooth areas are contaminated to embed the secret bits. Thus, the embedding noises caused by LSB alternations in the data hiding process are easily recognized by the human eye after these modified LSBs are enhanced.

To compare the ability to prevent from being discovered by visual attack between EA_LSBMR and the proposed approach MPBDH at the embedding rate of 0.8 bpp, the

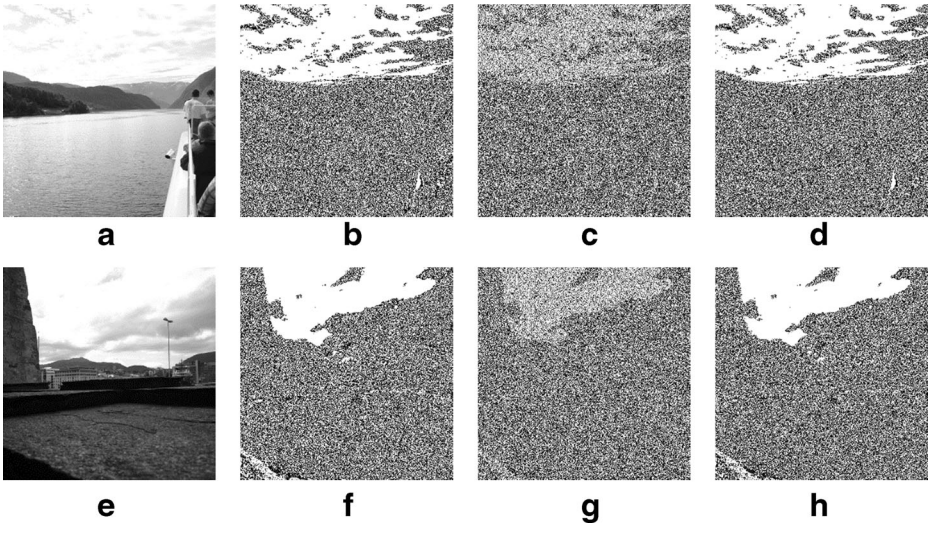

Fig. 8 The cover images (**a**, **e**) and their LSB planes (**b**, **f**), (**c**, **g**) LSB of the stego-images of EA_LSBMR and (**d**, **h**) LSB of the stego-images of the proposed method MPBDH with the same embedding rate of 0.8 bpp

cover images and their corresponding resulting LSB planes were presented in Fig. 8. As can be seen, the smooth regions in the stego images (Fig. 8(c–g)) were contaminated after data embedding. In contrast with the stego-images of EA_LSBMR, it is observed that the flat regions of the stego-images which are introduced by MPBDH are well preserved.

4.2.2 Ensemble classifier

The LSB Enhancement attack method is very effective at detecting the existence of data hidden in a selected stego-image that contains some smooth regions. Unfortunately, it is difficult to apply this attack method to stego-images containing high texture characteristic regions into which a secret message is embedded. Hence, a feature-based steganalysis algorithm Ensemble classifier [13] is implemented to evaluate the security performance of the proposed approach against statistical attack methods.

The ensemble classifier is designed with significantly lower training complexity and overall simplicity. These advantages of this approach open up the possibility of the steganalyst working with high dimensional cover models and larger training sets. Out-of-bag (OOB), an unbiased estimate of a real testing error, is an output of ensemble classifiers. The high value of OOB offers lower detectability under the ensemble classifier.

In this experiment, different splits of the BOSS image database were employed as cover images with embedding rates in the range of [0.10–0.40] bpp. The features of the cover images and their corresponding stages were then extracted by a SPAM feature extractor [18] which exports 686 features from an image. After that, the feature sets were used as input data for Ensemble classifier and OOB values were generated. The training and testing processes were conducted 20 times to achieve the average OOB values for each embedding rate.

The average obtained OOB values of the MPBDH and previous approaches are shown in Table 6 under several payloads. As can be seen from the table, the OOB values of the MPBDH are slightly higher than those of other methods at the payload of 0.10 bpp. For the higher embedding rates from 0.15 to 0.40 bpp, the OOB values of the proposed approach are relatively higher than those of other methods.

Table 6 Average OOB values of 10,000 stego-images produced by the proposed method (MPBDH) and previous approaches for embedding rates (from 0.10 bpp to 0.40 bpp)

Payload (bpp)	OOB				
	CBL	PRSA	EDSI	MPBDH	Improvement
0.10	0.232	0.218	0.206	0.310	7.86 %
0.15	0.092	0.110	0.200	0.273	8.13 %
0.20	0.028	0.066	0.104	0.236	8.35 %
0.25	0.012	0.042	0.096	0.209	8.67 %
0.30	0.004	0.054	0.094	0.192	8.59 %
0.35	0.004	0.082	0.136	0.148	8.76 %
0.40	0.002	0.048	0.072	0.111	8.95 %

Table 7 Average OOB values of 10,000 stego-images produced by the proposed method (MPBDH) and previous method EA_LSBMR for embedding rates (from 0.10 bpp to 0.40 bpp)

Payload (bpp)	OOB error		
	EA_LSBMR	MPBDH	Improvement
0.10	0.366	0.310	−15.30 %
0.20	0.338	0.236	−30.18 %
0.30	0.280	0.192	−31.43 %
0.40	0.216	0.111	−48.61 %

The last column of Table 6 illustrates the percentage of improvement of the proposed approach in comparison with EDSI (the method with steady obtained OOB values). The increase in these percentages indicates that MPBDH is outperforming the previous approaches (CBL, PRSA and EDSI), even when the payload is increased. Nevertheless, as shown in Table 7, the security against ensemble classifier of the proposed approach MPBDH is relatively lower than that of the previous method EA_LSBMR for different embedding rates.

The reason is that the EA_LSBMR employs LSB Matching Revisited (LSBMR) [15] method to hide a secret message. This method embeds a pair of message bits into pair of pixels, in which the LSB of the first pixel carries one bit, and the relationship of the two pixel values identify another bit of secret message. As a result, the modification rate of pixels is decreased to 0.375 bits per pixel in EA_LSBMR. In BDH method, which is used in the proposed approach MPBDH, the modification rate is only small when a big block of pixels is employed. However, in that case, the number of message bits can be hidden is reduced. Therefore, the ability to prevent being detected by ensemble classifier of the proposed approach MPBDH is lower than that of EA_LSBMR method. In addition, via employing random pixel pair selection scheme and the pixel is modified by adding +1 or −1 randomly, the security against statistical attacks (such as ensemble classifier) is further enhanced.

Nevertheless, the existence of an unseen message, which is embedded by EA_LSBMR, can be easily detected by the LSB enhancement attack (please refer to Section 4.2.1). Meanwhile, the ensemble classifier is possible only when a large number of cover images

Table 8 The average number of modifications (over 10,000 images) made by the proposed method (MPBDH) and previous approaches to embed a given message under various payloads

Payload (bpp)	Number of embedding changes				
	EDSI	PRSA	CBL	MPBDH	Reduction
0.10	13107	20906	13213	10223	22.00 %
0.15	19664	31499	19762	15357	21.90 %
0.20	26215	41895	26258	20469	21.92 %
0.25	32768	51803	34563	25561	21.99 %
0.30	39312	61303	39411	30656	22.02 %
0.35	45870	69941	46040	35754	22.05 %
0.40	52425	77615	55678	40865	22.05 %

and their corresponding stego-images are available to an opponent (attacker). This means that for a small amount of cover images and their corresponding stego-images, visual attack methods are more suitable to figure out the existence of hidden information in stego-images.

4.3 Embedding efficiency discussion

According to a formal definition of steganographic security given by Cachin [3], the detectability of data hiding in a stego object is influenced by many factors, such as the choice of stego object, the selection rule used to identify elements of the cover media, the type of embedding method that alternates the cover elements, and the number of embedding changes. Hence, the average number of embedding alternations, which were made by the MPBDH and previous approaches, is estimated and shown in Table 8 to illustrate the reduction in embedding change of the proposed approach in comparison with the previous methods.

The last column of this table presents the percentage reduction of the average number of modifications done by MPBDH in comparison with EDSI. It is obvious that the number of necessary modifications to hide a given message of the proposed approach is smaller than those of other methods. Thus, the proposed algorithm can employ adaptive adjustment to reduce the distortions caused by the embedding process. As a result, the perceptual quality to the stego-images introduced by the MPBDH is higher that of the previous approaches.

Moreover, a smaller number of embedding changes are made to hide the given message, the fewer the number of degradations of structural information in stego-images is introduced. In other words, the features in stego-images are almost as same as those in the cover images. As a result, the obtained PSNR and SSIM values of the proposed approach are higher than those of previous approaches for different embedding rates.

5 Conclusion

In this paper, we presented the MPBDH that obtains a high embedding capacity while maintaining appreciable visual quality. This is because in the proposed approach, more than one bit-plane of the block of pixels is used to hide secret bits. As a result, the capacity is higher than that of the existing block complexity based data hiding approaches.

Furthermore, employing the BDH, in which the necessary number of embedding changes is minimized in comparison with other methods, causes the distortions by the data hiding process to be reduced.

Additionally, by selecting a high texture characteristic block of pixels in which to embed the given secret message, even at the high embedding rates, the smooth regions are kept the same as in the cover image. This leads to a reduction in the possibility of being detected by visual attack methods. The experiment, in which 10,000 natural images were used and different steganalysis methods performed, shows that both the visual quality and the security of the proposed scheme are significantly improved when compared with block complexity and pixel complexity based approaches.

As future work, we will focus on investigating whether the technique used to guarantee the complexity of the block is higher than the threshold before embedding. This approach helps to identify whether the considered block contains hidden data or not without using the indicator bit. In addition, the capacity is also enhanced as well when the bits that are used to mark the block of pixels are employed to hide the secret bits.

Acknowledgments We gratefully acknowledge Department of Computer Science for financial support and Khon Kaen University for their assistance.

References

1. (1997) SIPI Image Database. http://sipi.usc.edu/database/. Accessed 24 Jan 2015
2. Battisti F, Cancellaro M, Boato G et al (2009) Joint watermarking and encryption of color images in the fibonacci-haar domain. EURASIP J Adv Signal Process 2009:938515. doi:10.1155/2009/938515
3. Cachin C (1998) An information-theoretic model for steganography. inf. hiding. Springer, Berlin, pp 306–318
4. Chang K-C, Huang PS, Te-Ming Tu, Chien-Ping Chang (2007) Adaptive image steganographic scheme based on Tri-way Pixel-Value Differencing. IEEE, 1165–1170
5. Daemen J, Rijmen V (2002) The design of Rijndael AES - the advanced encryption standard. Springer, Berlin
6. Fridrich J, Goljan M, Du R (2001) Reliable detection of LSB steganography in color and grayscale images. ACM Press, 27
7. Hashemi Pour A (2012) A new steganography method based on the complex pixels. J Inf Secur 03:202–208. doi:10.4236/jis.2012.33025
8. Hirohisa H (2002) A data embedding method using BPCS principle with new complexity measures. Proc Pac. Rim Workshop Digit. Steganography. 30–47
9. Hong W (2012) Human visual system based data embedding method using quadtree partitioning. Signal Process Image Commun 27:1123–1133. doi:10.1016/j.image.2012.09.002
10. Jung K-H, Yoo K-Y (2014) Data hiding using edge detector for scalable images. Multimed Tools Appl 71: 1455–1468. doi:10.1007/s11042-012-1293-8
11. Kawaguchi E, Eason RO (1998) Principle and applications of BPCS-steganography
12. Ker AD (2005) Steganalysis of LSB matching in grayscale images. IEEE Signal Process Lett 441–444. doi:10.1109/LSP.2005.847889
13. Kodovsky J, Fridrich J, Holub V (2012) Ensemble classifiers for steganalysis of digital media. IEEE Trans Inf Forens Secur 7:432–444. doi:10.1109/TIFS.2011.2175919
14. Luo W, Huang F, Huang J (2010) Edge adaptive image steganography based on lsb matching revisited. IEEE Trans Inf Forens Secur 5:201–214. doi:10.1109/TIFS.2010.2041812
15. Mielikainen J (2006) LSB matching revisited. IEEE Signal Process Lett 13:285–287. doi:10.1109/LSP.2006.870357
16. Muñoz A (2007) StegSecret. A simple steganalysis tool. http://stegsecret.sourceforge.net/. Accessed 24 Jan 2015
17. Nguyen TD, Arch-int S, Arch-int N (2014) A novel secure block data-hiding algorithm using cellular automata to enhance the performance of JPEG steganography. Multimed Tools Appl. doi:10.1007/s11042-014-1877-6
18. Pevny T, Bas P, Fridrich J (2010) Steganalysis by subtractive pixel adjacency matrix. IEEE Trans Inf Forens Secur 5:215–224. doi:10.1109/TIFS.2010.2045842
19. Qi H, Zheng D, Zhao J (2008) Human visual system based adaptive digital image watermarking. Signal Process 88:174–188. doi:10.1016/j.sigpro.2007.07.020
20. Rivest RL, Shamir A, Adleman LM (1983) Cryptographic communications system and method. Google Patents
21. Rose NJ (2001) Hilbert-type space-filling curves
22. Sabeti V, Samavi S, Shirani S (2013) An adaptive LSB matching steganography based on octonary complexity measure. Multimed Tools Appl 64:777–793. doi:10.1007/s11042-011-0975-y
23. Sur A, Ramanathan V, Mukherjee J (2014) Pixel rearrangement based statistical restoration scheme reducing embedding noise. Multimed Tools Appl 68:805–825. doi:10.1007/s11042-012-1078-0

24. Tomáš Pevný, Tomáš Filler, Patrick Bas (2013) Break Our Steganography System. s. Accessed 24 Jan 2015
25. Voloshynovskiy S, Herrigel A, Baumgaertner N, Pun T (2000) A Stochastic Approach to Content Adaptive Digital Image Watermarking. In: Pfitzmann A (ed) Inf. Hiding. Springer, Berlin, pp 211–236
26. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error measurement to structural similarity. IEEE Trans Image Process 13:600–612
27. Westfeld A, Pfitzmann A (2000) Attacks on Steganographic Systems. In: Pfitzmann A (ed) Inf. Hiding. Springer, Berlin, pp 61–76

Tuan Duc Nguyen received the M. S degree from Le Qui Don Technical University, Vietnam, in 2008. He is currently a Ph.D student in Department of Computer Science, Faculty of Science, Khon Kaen University, Thailand. His research interests are cryptography, steganography and steganalysis.

Somjit Arch-int received the PhD degree in computer science from the Asian Institute of Technology in 2002. He is currently an associate professor in the Department of Computer Science, Khon Kaen University, Thailand. His previous experiences include the development of several industry systems and consulting activities. His research interests are business component-based software development, object-oriented metrics, ontology-based e-business modeling, knowledge-based representation, semantic information integration, data mining, and semantic Web. He is a member of IEEE Computer Society.

Ngamnij Arch-int received the PhD degree in computer science from Chulalongkorn University, Thailand in 2003. She is currently an associate professor in the Department of Computer Science at Khon Kaen University, Thailand. Her research interests include the semantic web, web services, semantic web services, and heterogeneous information integration. Contact her at ngamnij@kku.ac.th