Expert Systems With Applications 46 (2016) 293–306

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A steganography embedding method based on edge identification and XOR coding

Hayat Al-Dmour*, Ahmed Al-Ani

School of Electrical, Mechanical and Mechatronic Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW 2007, Australia

ARTICLE INFO

Keywords:
Steganography
Edge detection
Human Visual System (HVS)
XOR coding
IWT

ABSTRACT

In this paper, we present a novel image steganography algorithm that combines the strengths of edge detection and XOR coding, to conceal a secret message either in the spatial domain or an Integer Wavelet Transform (IWT) based transform domain of the cover image. Edge detection enables the identification of sharp edges in the cover image that when embedding in would cause less degradation to the image quality compared to embedding in a pre-specified set of pixels that do not differentiate between sharp and smooth areas. This is motivated by the fact that the human visual system (HVS) is less sensitive to changes in sharp contrast areas compared to uniform areas of the image. The edge detection method presented here is capable of estimating the exact edge intensities for both the cover and stego images (before and after embedding the message), which is essential when extracting the message. The XOR coding, on the other hand, is a simple, yet effective, process that helps in reducing differences between the cover and stego images. In order to embed three secret message bits, the algorithm requires four bits of the cover image, but due to the coding mechanism, no more than two of the four bits will be changed when producing the stego image. The proposed method utilizes the sharpest regions of the image first and then gradually moves to the less sharp regions. Experimental results demonstrate that the proposed method has achieved better imperceptibility results than other popular steganography methods. Furthermore, when applying a textural feature steganalytic algorithm to differentiate between cover and stego images produced using various embedding rates, the proposed method maintained a good level of security compared to other steganography methods.

1. Introduction

Since the early stages of the human civilization, there has been an increased interest in information security, particularly the protection and privacy of communications (Pal & Pramanik, 2013). In modern societies, the excessive use of electronic data has made protection from malicious users more difficult (Grover & Mohapatra, 2013). Information hiding has emerged as an effective solution to this problem (Wu & Tsai, 2003; Wu, Lee, Tsai, Chu, & Chen, 2009).

Steganography is a kind of information hiding, in which a secret message is concealed within digital media (image, audio, video or text data) (Bassil, 2012; Cheddad, Condell, Curran, & Mc Kevitt, 2010). This property distinguishes steganography from other information security techniques (Modi, Islam, & Gupta, 2013). For instance, in cryptography, the message that needs to be transferred is encrypted to prevent intruders from understanding it. Hence, people can recognize the existence of the message, however it cannot be understood without decryption (Bassil, 2012; Cheddad et al., 2010; Verma, 2011).

As opposed to data concealing, steganalysis was initially designed to distinguish whether a given digital media has a secret message embedded in it. Moreover, some steganalysis methods may determine the type of steganography technique or estimate the length of the secret message (Li, He, Huang, & Shi, 2011). In term of security measurement, steganalysis has been utilized to evaluate the efficiency of steganography techniques from a security point of view (Geetha, Ishwarya, & Kamaraj, 2010). Steganalysis methods can be performed either by using image processing operation or by implementing methods that analyze the statistical features of the stego image structure, such as first order statistics (histogram) or second order statistics (correlations between pixels) (Cheddad et al., 2010). Ziou and Jafari suggested five requirements for steganalysis methods: (1) detection of the existence or absence of an embedded message in a given image, (2) identification of the steganographic method that have been used to hide the secret message, (3) approximation of the hidden message length or location and (4) extraction of the secret message (Ziou & Jafari, 2014).

* Corresponding author. Tel.: +61-426401478.
E-mail addresses: HayatShahir.T.Al-Dmour@student.uts.edu.au, HayatDmour@gmail.com (H. Al-Dmour), Ahmed.Al-Ani@uts.edu.au (A. Al-Ani).

http://dx.doi.org/10.1016/j.eswa.2015.10.024
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

Table 1
Differentiation between image steganography schemes in spatial and transform doamins.

Domain	Advantages	Disadvantages
Spatial Domain	High embedding capacity Shorter computational time High controllable imperceptibility	Vulnerable to geometric attacks.
Transform Domain	Robustness against attacks such as Geometric attacks and compression	High computational time Limited embedding capacity Lower controllable imperceptibility

Recently, steganography has grown in popularity (Grover & Mohapatra, 2013). Digital images have particularly been the focus of many researchers because of their high degree of redundancy (stored with an accuracy far greater than necessary for the data's use and display (Morkel, Eloff, & Olivier, 2005)). Also, using images as cover will not create any suspicion due to their widespread use on the Internet (Cheddad et al., 2010). Three major requirements should be considered when evaluating a steganography scheme: data embedding rate, imperceptibility, and robustness (Bassil, 2012; Chen, Chang, & Le, 2010). These three evaluation factors are needed as it is important for steganography techniques to have a high capacity and to be undetectable (Chan & Chang, 2010; Wu et al., 2009). The steganography terminology is listed below.

- Cover image (C): carrier medium used to hide the message.
- Stego image (S): output of the embedding process.
- Message (M): secret message to be hidden within the cover image.
- Key (K): used to encrypt the message before embedding (optional).
- Embedding Process (Em): the process of generating S by hiding M into C.
- Extraction Process (Ex): the process of retrieving M from S.

Mathematically, the embedding (or concealing) process can be represented as: $S = Em(C, M, K)$, and the extraction process as: $\acute{M} = Ex(S, K)$. The extraction process should be reversible to the embedding process. Hence, $Ex(Em(C,\ M,\ K),\ K)$ should be equal to M (or $\acute{M} = M$).

Numerous steganography methods have been proposed in the literature (Cheddad et al., 2010; Luo, Huang, & Huang, 2010; Verma, 2011). These methods can be partitioned based on the embedding domain; spatial and transform (Bassil, 2012; Grover & Mohapatra, 2013; Ioannidou, Halkidis, & Stephanides, 2012). In spatial domain steganography methods, M is directly embedded in the pixels of C. Two of the most famous steganography techniques are the Least Significant Bit (LSB) and Pixel Value Differencing (PVD). Transform domain steganography methods, on the other hand, transform C into another domain by performing one or more transforms, such as Discrete Transform Domain (DCT), Discrete Wavelet Transform (DWT), or Singular Value Decomposition (SVD), and then embed M by modifying the transformed coefficient values (Cheddad et al., 2010; Kanan & Nazeri, 2014). Table 1 presents the difference between image steganography in spatial and transform domains in term of embedding capacity, imperceptibility and robustness (Bandyopadhyay, Dasgupta, Mandal, & Dutta, 2014; Ghebleh & Kanso, 2014; Hussain & Hussain, 2013).

The Least Significant Bit (LSB) hiding is the most common methodology to implement steganography (Morkel et al., 2005). LSB-based steganography is based on manipulating the LSBs of some or all pixels of the cover image to embed the message and can be classified into two main types; LSB replacement (LSBR) and LSB matching (LSBM) (Luo et al., 2010; Zhu, Zhang, & Wan, 2013). LSB-based steganography methods allow the concealment of a large amount of data. Another advantage of LSB steganography is the simple extraction process (Luo et al., 2010). Usually, a pseudo random number generator is used to improve security of LSB steganography by spreading the message on the cover randomly (Luo et al., 2010).

In order to enhance the embedding efficiency, coding methods (mainly matrix encoding) have been introduced with the aim of minimizing the modifications created by embedding the message (Crandall, 1998; Hou, Lu, Tsai, & Tzeng, 2011).

In this paper, we propose a functional and simple image steganography method that is based on identifying edge locations on the cover image and incorporates an XOR coding function. The XOR function, which has a lower computation complexity compared to other matrix encoding methods, adds some security and reduces the distortion caused by embedding the message. Embedding in both spatial and wavelet transformed domains has been implemented.

The rest of this paper is organized as follows. Section 2 provides a brief introduction to wavelet transform. Section 3 describes some of the existing steganography methods and examines their strengths and weaknesses. Details of the proposed method are presented in Section 4. Section 5 presents the experimental results, and the conclusion is given in section 6.

2. Wavelet transform

Transform domain embedding methods provide a higher level of robustness, particularly when applying some image processing operations, compared to spatial domain methods. One of the most popular transforms is the Discrete Wavelet Transform (DWT) (Baby, Thomas, Augustine, George, & Michael, 2015; Thanikaiselvan et al., 2014). The Wavelet transform requires less computational cost compared to DCT and FFT (Fourier Transform) and offers sub-representations of the image that can be considered related to how the human visual system (HVS) perceives images. Generally, the wavelet transform allows embedding data in high frequency regions where the HVS cannot distinguish modifications compared to uniform regions with low frequency (Sharma & Swami, 2013). When DWT is performed to an image it is divided into 4 sub-bands: Low–Low (LL), Low–High (LH), High–Low (HL) and High–High (HH) frequency sub-bands, as shown in Fig. 1. The low frequency sub-band represents coarse information of pixels, while the high frequency sub-bands represent the edge information (Sharma & Swami, 2013). Hiding Information in the high frequency sub-bands (LH, HL, and HH) increases the robustness and ensures the visual quality, where the HVS is less sensitive to modifications in these sub-bands. The Integer Wavelet Transform (IWT) maps integers to integers and allows the construction of lossless compression to exactly retrieve the original data (Thanikaiselvan et al., 2014).

3. Related work

Imperceptibility is an essential requirement for steganography techniques, which reflects the ability of these techniques in maintaining the visual quality of the produced stego images. It is well-known that the HVS is less sensitive to changes in sharp areas of images compared to smooth areas. The first steganography method designed based on this fact was the Pixel-value differencing (PVD), which attempts to embed into sharp areas. The original PVD algorithm introduced by Wu and Tsai (2003) converts the 2D image into a

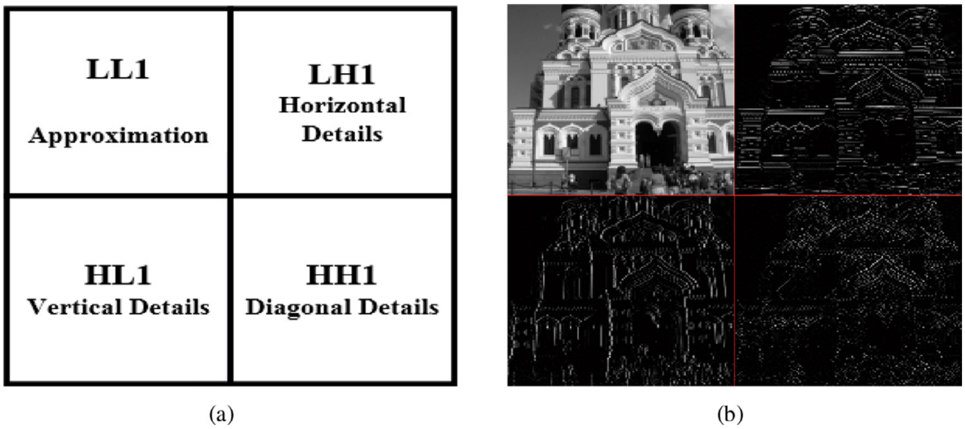


Fig. 1. (a) DWT sub-bands and (b) An example of the First Level DWT decomposition.

1D vector. The number of bits that can be used for embedding in each pixel is calculated based on the difference between that pixel and its neighbour. Thus, more bits are to be embedded in a pixel if its grey level is noticeably different from that of its neighboring pixel. This method, however, only considers differences in one dimension (either horizontal or vertical), which does not guarantee that all edges are identified. This implementation can also increase the possibility of detecting the message by tracing pixels in every block. As it will create noticeable change in the adjacent pins in the histogram (Zhang & Wang, 2004). In other words, it changes the relationship between consecutive pixels, and hence may cause distortion in the histogram of the stego-image. Because of this limitation, many revisions have been introduced to the original PVD algorithm, such as (Luo et al., 2010). An alternative approach is the utilization of edge detection algorithms, which has recently received an increased interest from the steganography community (Zhang & Wang, 2004). Since the intensity of edge pixels is either higher or lower than their neighboring pixels, edge pixels might be distinguished as noisy pixels. Because of their high variations in statistical characteristics, edge regions can be considered as the best regions to conceal the secret message compared to any other region of the image.

3.1. Utilization of edge detection in steganography

The utilization of edge detection in image steganography has been considered by a number of researchers. Due to sensitivity of the human eye to changes in smooth areas of the image compared to sharp contrast areas, it is logical to focus on sharp edges when embedding the secret message. However, the main obstacle to applying traditional edge detection methods in image steganography is the correct identification of edge pixels in the stego image S that need to exactly match the original edge pixels in the cover image C. This problem arises from the fact that the embedding process introduces minor changes to the stego image, which may make the produced stego image not identical with the cover image, and this can affect the message extraction process. Some of the existing edge-based steganography methods suggested certain solutions to overcome this problem. Chen et al. (2010) applied a hybrid edge detection method to conceal the message. An edge image is created by performing Canny and fuzzy edge detection methods. The cover is then distributed into blocks of n pixels. The first pixel of each block is changed to represent the status of $(n-1)$ pixels if it is considered as edge pixel. LSB technique is used to embed x bits into non-edge pixels and y bits into edge pixels. The main drawback of this method is the unwanted modification that are created in the stego image because the method replaces $(n-1)$ bits from the first pixel of each block.

Li, Luo, Li, and Fang (2009) introduced a spatial color image steganography based on Sobel operators. Sobel edge detection was performed on one R, G or B channel of the cover image. Embedding locations are chosen based on the largest number of gradients among R, G and B planes. The LSB of corresponding pixels in different planes are altered to conceal data. Embedding capacity is improved by repeating these phases many times until the secret message is embedded. Finally, the separate planes are integrated to form the stego-image. However, this process does not guarantee a high embedding rate and the data extraction is sometimes incorrect because the method may embed data into the LSB pixels more than one time. Bassil (2012) proposed a color image steganography method based on canny edge detection to select the embedding location and LSB techniques to hide the message into the cover. The embedding rate is increased by this method because it hides 3 LSBs in every pixel detected by the canny method. However, this method does not introduce any solution to correctly identifying the same edge pixels.

Luo et al. (2010) designed an edge adaptive LSB Matching Revisited (EALSB-MR) algorithm. The method discovers vertical and horizontal edges in an adaptive way. It searches for edge regions by calculating the difference between consecutive pixels. The selection of regions depends on the secret message length and is verified by a threshold value. The method uses horizontal and vertical edges by dividing the image into blocks then rotating each block by a random angle, however, this process could destroy the relationship between vertical/horizontal pixels (Modi et al., 2013).

Modi et al. (2013) introduced a color image steganography based on the Canny edge detection method and LSBM to hide 2 LSBs on the edge pixels. Canny method is applied on one channel only (R, G, or B). Then the pixels in the other two channels corresponding to the edge pixels are selected for embedding. The main limitation of this method is the low embedding capacity, where the payload is 0.083 bpp of the edge pixel of the color image. Also, it cannot be utilized for grayscale images.

3.2. Utilization of the coding theory

Enhancing the embedding efficiency has been the focus of many steganography algorithms, as minimizing the amount of changes in the image when embedding (embedding rate) will enable the embedding of bigger messages. Matrix embedding was introduced by Crandall (1998) to enhance the hiding efficiency through minimizing the difference between the cover image and the stego image. Crandall's method utilizes the XOR function to conceal 2 bits of message into a block of 3 pixels. This procedure has an embedding rate

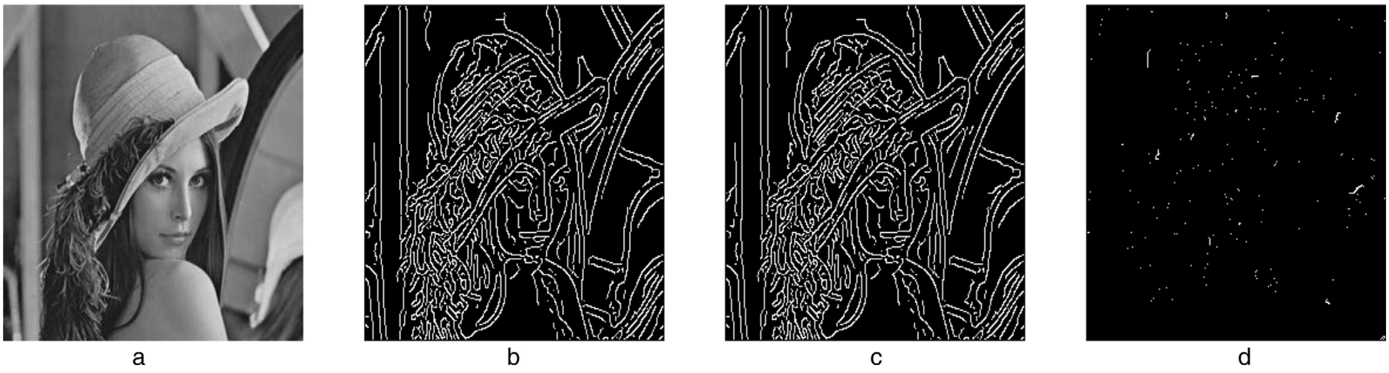


Fig. 2. (a) Cover image, (b) Edge pixels in a cover image using Canny method, (c) Edge pixels in a stego image using Canny method, and (d) Difference between edge pixels in the cover and stego images.

of 67% and a change rate of 25%. The F5 steganography algorithm, proposed by Westfeld (2001), is the first execution of matrix encoding to increase the capacity of embedding data as well as to minimize the change of DCT coefficients. This method has become well-known because it integrated the Hamming code with the transform domain implementation, which can embed k bits of the secret data in $2^k - 1$ cover bits by changing at most one bit only. As a result, this method has a limited embedding capacity, for example when $k = 3$, the method only embeds 3 bits in every 7 bits of the cover image. Another limitation of matrix embedding is the computational cost, as it requires matrix multiplication (Westfeld, 2001).

Hou et al. (2011) introduced an approach called tree based parity check (TBPC) that uses a tree structure to enhance the embedding efficiency by reducing deformation on the cover object. They proposed a strategy of majority voting for TBPC and argued that this strategy inherited the efficiency of the TBPC method and produced the least deformation. Similar to some of the other coding methods, the drawback of this method is the high computational cost, especially for trees that have multiple levels. The method can hide 2^n bits in a binary tree of n levels. For example, if the binary tree has 2 levels, then it hides 4 secret bits into 7 pixels.

3.3. *Utilization of integer wavelet transform*

Reddy and Raja (2011) proposed a wavelet based on LSB steganography method in which the original image is divided into 4 × 4 pixels and DWT/IWT is performed on each block to form 2 × 2 sub-bands block. The embedding process is performed on the HH sub-band of DWT/IWT. The 2 × 2 HH coefficients are modified to carry bit pairs of the secret data using identity matrix to produce the stego image. For the extraction process, the key and the stego image are used to retrieve payload. The authors claimed that this method cannot be discovered by some steganalysis methods such as Chi-square and pair of values methods.

Kanan and Nazeri (2014), proposed a steganographic technique based on IWT and genetic algorithm where IWT is utilized to avoid floating point precision problem of the wavelet filter. It embeds the secret data in the IWT coefficients by using a mapping function based on a Genetic Algorithm implementation of an 8 × 8 block of the cover image. The Optimal Pixel Adjustment Process (OPAP) is performed after hiding the secret data to reduce the difference error between the cover and stego images and to improve the embedding payload with good image quality.

The next section presents our proposed steganography embedding algorithm with its two implementations; one for the spatial domain while the other for the wavelet transform domain. We also present a simple, yet effective, coding process that is not as computationally demanding as most of the existing coding methods.

4. The proposed method

4.1. *The spatial domain algorithm*

4.1.1. *Identification of edges*

As mentioned earlier, the human visual system is less tangible to changes in image areas that contain edges and sharp transitions in comparison to smooth areas. Accordingly, it is logical to conceal the message in edge areas in order for the steganography algorithm to have a good imperceptibility.

The edge image generated by traditional edge detection methods is usually sensitive to changes in the original gray image, even if the changes are minor or not significant. This property limits the utilization of edge detection in steganography, as concealing the message would introduce some changes to the original image. Thus, embedding in pixels identified by one of the existing edge detection methods, such as Canny, cannot guarantee the identification of the exact edge intensities for the cover and stego images. Fig. 2(a) and (b) shows the cover image and the corresponding edge pixels which are identified by applying Canny edge detection. Edge pixels of the stego image produced after embedding a message of length 26214 bits is shown in Fig. 2(c). Fig. 2(d) shows the difference between the two edge images, which indicate that edge pixels in the cover and stego images are not identical.

We propose here a simple new way to discover the edge (sharp) regions of the cover image, such that the two edge images generated using the original cover image and the stego image are identical. This will enable the correct extraction of the concealed message from the stego image. The algorithm starts by dividing the image into non-overlapping blocks that would be individually evaluated for inclusion of edges. The key idea behind preserving the same edge image is not to embed in the pixels that are used to calculate the edge strength, which are the outer pixels of the block.

We will first explain how to embed one bit per each of the center pixels, and then expand that to n bits per pixel. Below are the detailed implementation steps.

Inputs: Cover image (C), block size ($n \times n$, which is expected here to be 3 × 3), threshold (Th ranges between 4 and 96)

Output: edge image with edge magnitude (E)

Step 1 : Divide the image C into non-overlapping blocks of the size $n \times n$.

Step 2 : Compute the absolute mean difference between the left and right columns of the block (magnitude of vertical edge). Repeat for horizontal, first diagonal and second diagonal edges. Fig. 3(a) shows the specific pixels used to calculate the edges for the 3 × 3 block.

Step 3 : Find the maximum of the four values and assign it to e. If $e > Th$, then the block is considered to be an edge block, otherwise

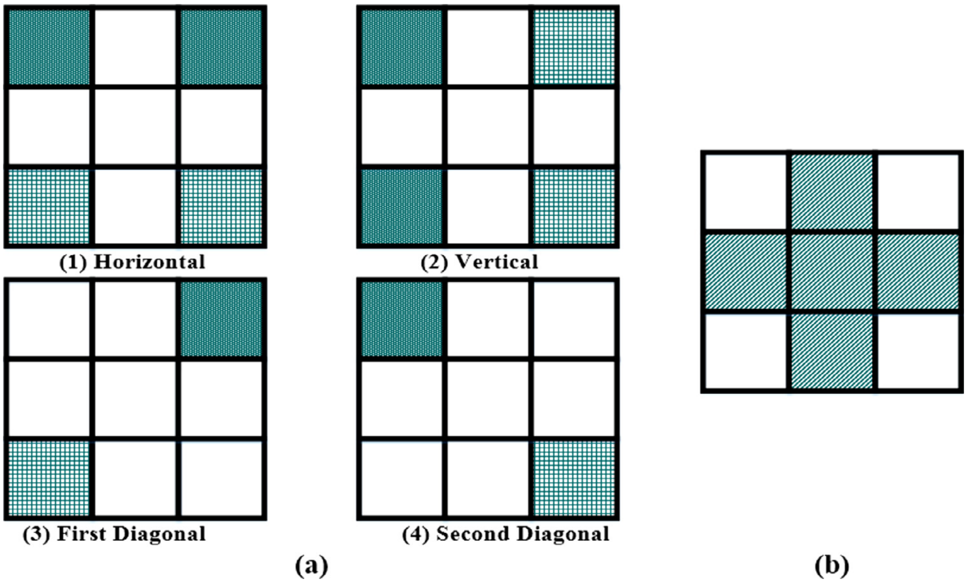


Fig. 3. (a) 3 × 3 block edges and (b) Selected pixels for embedding 3 × 3 block.

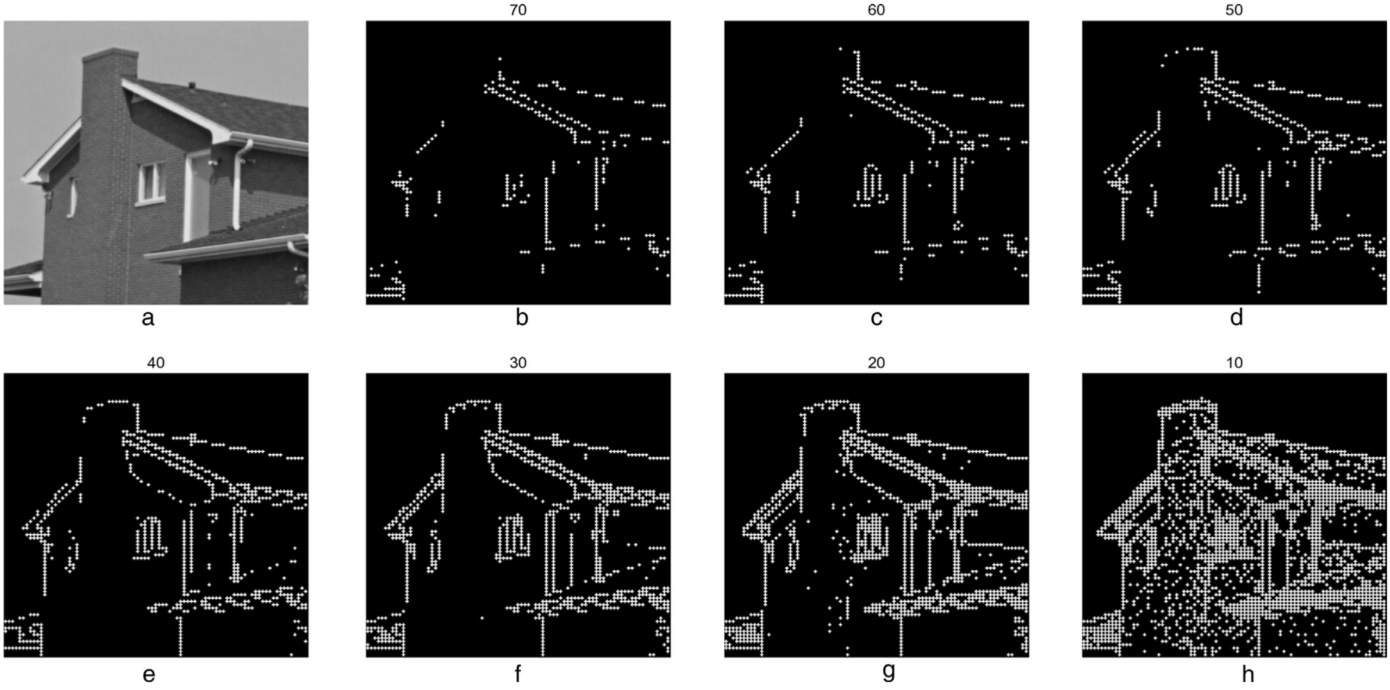


Fig. 4. (a) Input image, (b) edge image using $Th = 70$, (c) edge image using $Th = 60$, (d) edge image using $Th = 50$, (e) edge image using $Th = 40$, (f) edge image using $Th = 30$, (g) edge image using $Th = 20$, (h) edge image using $Th = 10$.

it is not an edge block. Construct E that contains the calculated e value of each of the edge blocks, and 0 for non-edge blocks. A binary edge image can also be constructed, which contains 1 for edge blocks and 0 for non-edge blocks.

Step 4 : For the edge blocks, embed in the shaded 5 pixels as shown in Fig. 3(b).

In order to evaluate the obtained binary edge image of the proposed algorithm, we considered the gray image shown in Fig. 4(a) and used different values of threshold in constructing binary edge images using a block size of 3 × 3, as shown in Fig. 4(b)–(h). The edge images indicate the ability of this method in detecting edges with an acceptable accuracy. Out of the nine pixels of the block, the five pixels shown in Fig. 3(b) will be used for embedding if the block is identified as an edge block. Thus, the four corner pixels that are used for estimating the edge strength will remain unchanged after embedding. This guarantees each block in the cover image to have the same edge strength as its counter part in the stego image.

4.1.2. Message embedding

The flow diagram of our proposed method is illustrated in Fig. 5. The data embedding process begins with reading the cover image and the secret message. A high threshold (96) is initially considered, which is then adjusted based on the number of pixels needed for embedding (identified by the generated binary edge image) and the message length, according to the following condition:

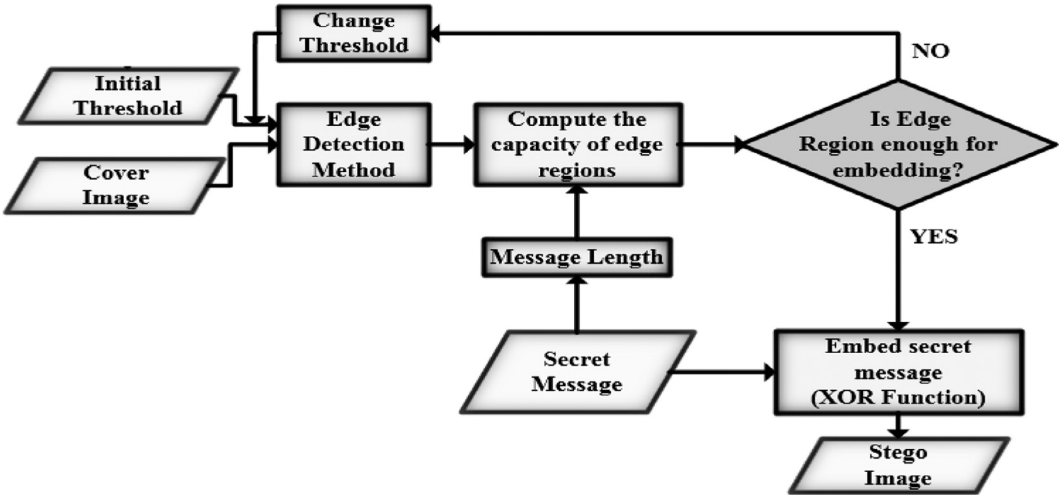


Fig. 5. Data embedding process in the spatial domain.

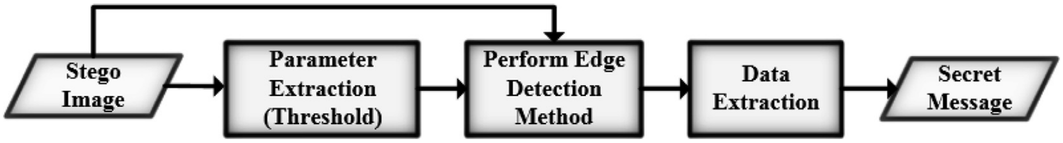

Stego Image
Parameter Extraction (Threshold)
Perform Edge Detection Method
Data Extraction
Secret Message

Fig. 6. Data extraction process in the spatial domain.

Table 2
Embedding conditions.

Condition			Action to be taken
$m_1 = k_1$	$m_2 = k_2$	$m_3 = k_3$	No change required
$m_1 = k_1$	$m_2 = k_2$	$m_3 \neq k_3$	Complement p_3 and p_4
$m_1 = k_1$	$m_2 \neq k_2$	$m_3 = k_3$	Complement p_4
$m_1 = k_1$	$m_2 \neq k_2$	$m_3 \neq k_3$	Complement p_3
$m_1 \neq k_1$	$m_2 = k_2$	$m_3 = k_3$	Complement p_2
$m_1 \neq k_1$	$m_2 = k_2$	$m_3 \neq k_3$	Complement p_1
$m_1 \neq k_1$	$m_2 \neq k_2$	$m_3 = k_3$	Complement p_2 and p_4
$m_1 \neq k_1$	$m_2 \neq k_2$	$m_3 \neq k_3$	Complement p_1 and p_4

*For the given threshold value, if (no. of edge pixels ≥ (4 * Message Length)/3)) then the discovered area is enough to embed the secret message.*

The embedding process is performed on the detected edge locations using the proposed XOR coding. This method partitions the index table into groups of four pixels and encodes three message bits into the pixels of each group. The XOR operation ensures that the secret message is concealed into the cover with minimum number of pixel changes. Thus, the three secret bits m_1, m_2, and m_3 are embedded in the four LSBs p_1, p_2, p_3, and p_4 (one bit for each edge pixel) according to the following procedure:

1. Perform the following three XOR operations
   $k_1 = p_1 \oplus p_2$
   $k_2 = p_3 \oplus p_4$
   $k_3 = p_1 \oplus p_3$
2. To embed the three secret bits m_1, m_2, and m_3, the three calculated bits k_1, k_2 and k_3 are compared with the secret message bits m_1, m_2, and m_3. The result of this comparison, which can take one of eight possibilities, determines which of the four bits p_1, p_2, p_3, and p_4 have to be modified, as shown in Table 2. We will refer to the new four bits of the stego image as q_1, q_2, q_3, and q_4. The table indicates that embedding 3 message bits into 4 cover bits will cause an average modification of 1.25 bits.
3. The threshold value should also be embedded, as it is needed by the extraction process. In this algorithm, the threshold value is embedded into the last pixel of the cover image.

4.1.3. *Message extraction*

The extraction process is easier and faster than the embedding process. Fig. 6 represents the flow diagram of the extraction process. It starts by retrieving the threshold value. The edge blocks of the stego image are then identified using the retrieved threshold, which will return the same edge image as the one obtained using the cover image. This will be followed by dividing the LSBs of the edge pixels into groups of four. Finally, for each of the four stego edge bits q_1, q_2, q_3, and q_4 the XOR operations listed below are used to retrieve three message bits m_1, m_2, and m_3

$m_1 = q_1 \oplus q_2$
$m_2 = q_3 \oplus q_4$
$m_3 = q_1 \oplus q_3$

When considering any combination of m_1, m_2, m_3, p_1, p_2, p_3, and p_4 to verify the embedding and extraction processes, one can find that the extraction process truly restores the original message.

4.1.4. *Embedding and extraction of* n *bits per pixel*

In order to improve the embedding capacity, we present here an extension of our 1 bpp algorithm to embed *n* bits in each edge pixel. The value of *n* is to be determined based on the edge mean value of each block. Thus, strong edges will enable the embedding of more bits than the less strong ones. Hence, unlike the embedding of one bit per pixel that only considers the existence of an edge in a block, this algorithm utilizes the edge strength of each block, *e*. Embedding *n* bits per pixel, where *n* varies from one block to another, may also improve the security of the message, as in this case *n* needs to be correctly calculated for each block in order to successfully reveal the message.

The data hiding process begins with reading the cover image and the secret message. The new edge detection is then applied to produce the edge strength, *e*, of each block. In order to specify the number of bits to embed, *n*, the edge pixels are classified into five groups (G_1, G_2, G_3, G_4 and G_5) based on the edge strength, *e*, of the block.

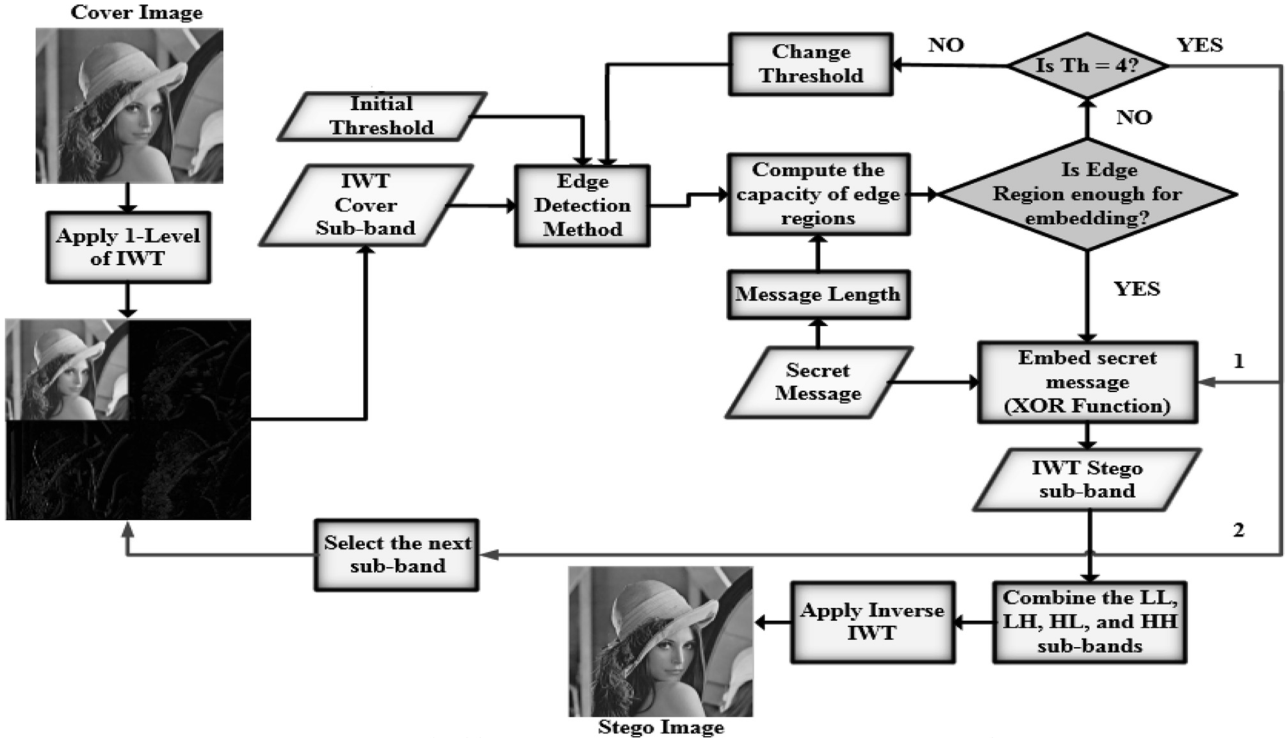


Fig. 7. Data embedding process in the Integer Wavelet Transform domain.

Table 3
Number of bits can be utilized from each edge pixel according to the group it belongs to.

Group	Group 1	Group 2	Group 3	Group 4	Group 5
n (bpp)	1	2	3	3	3
Range	[4, 7]	[8, 15]	[16, 31]	[32, 63]	[64, 255]

Table 3 lists the range of each of the five groups, and Eq. 1 determines the length of the message that can be embedded into the edge bits. If the identified edge pixels are not enough for embedding the whole message, then adjust the threshold and repeat the process until the actual length of the message satisfies Eq. 1.

$$(4 \times Message\,Length)/3) \le No.\,of\,Edge\,Bits[(3\times G_5\ \ pixel) + (3\times G_4\ \ pixel) + (3\times G_3\ \ pixel) + (2 \times G_2 pixel) + (1 \times G_1 pixel)] \qquad (1)$$

where the multiplier of each term represents the number of secret bits to embed in the edge pixel based on the group it belongs to.

The embedding process starts with the G_5 pixels and then moves to the remaining groups, where after it completes G_4 it moves to G_3 then G_2 and finally G_1 to embed 3, 3, 3, 2 and 1 bits in each of the corresponding edge pixels of these groups.

The retrieving process starts with performing the edge detection algorithm described earlier to get the edge strength, which would be used to categorize the edge blocks into group. Then, the XOR extraction operations are applied to extract three bits from the G_5 pixels, and then it respectively considers the G_4, G_3, G_2 and finally G_1 pixels to extract the corresponding number of bits from each of them.

4.2. Embedding and extraction of n bits per coefficient of the Integer Wavelet Transform Domain

The flow diagram of our proposed Integer Wavelet Transform (IWT) based embedding is illustrated in Fig. 7. The process starts by converting the cover image to the frequency domain using IWT. Since the HVS is sensitive to small modification into the lower frequency band compared to the higher frequency, the secret data is embedded only in the high frequency sub-bands of the IWT domain to achieve a high robustness and imperceptibility results. In other words, data hiding is carried out in the three sub-bands HH, LH and HL (the LL sub-band is excluded). Similar to the spatial domain embedding, the XOR operation is also utilized here.

The embedding process begins with HH sub-band and identify the edge coefficients to start embedding with the strongest edges to the weakest edges. If the HH sub-band is not enough to embed the secret message, then the process moves to the LH sub-band, and then to the HL sub-band.

The implementation of the embedding process is explained in the following steps:

1. Read the cover image and the secret message.
2. Apply the First-Level of IWT on the cover image to decompose the cover image into four sub-bands (LL, HL, LH and HH).
3. Identify edge regions in the high frequency sub-bands (LL, HL, LH and HH). To increase the embedding payload of the wavelet transform method, *n* LSB from each edge coefficients are utilized in embedding. A higher threshold value (*Th*) is initialized, which is then decreased based on the number of coefficients needed for embedding and the message length. To identify the edge regions, HH sub-band is divided into non-overlapping blocks of 3 × 3 coefficients as shown in Fig. 3(b). For each block, the average value (*avg*) of the four non-shaded coefficients $(P_{i-1,j-1}, P_{i-1,j+1}, P_{i+1,j-1}, P_{i+1,j+1})$ is calculated. Finally, if the average value (avg) ≥ Threshold, the block is selected for embedding.
4. Arrange the edge coefficients into five groups, as shown in Table 3. According to Eq. 1, if there are enough coefficients to embed the secret message, then embedding process is performed using XOR operation. Otherwise, repeat step 3 after adjusting the threshold value. This process is repeated on the other two sub-bands (LH

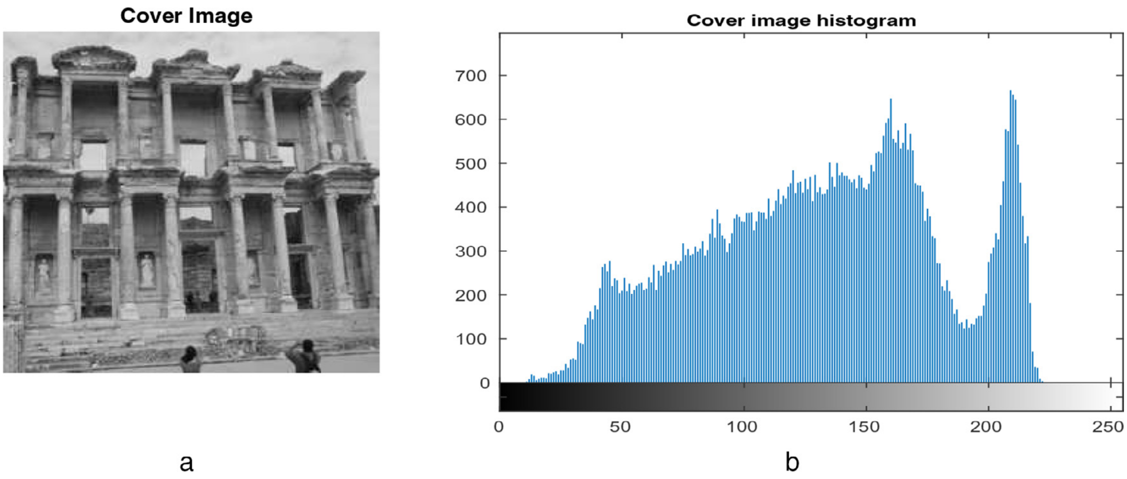


Fig. 8. (a) Cover image 512 × 512 and (b) Cover image histogram.

and HL) until finding enough area for embedding the whole message.

The extraction process begins with retrieving the threshold value to apply the edge detection method. Edge detection method is performed on the high frequency sub-bands by dividing the sub-band into non-overlapping blocks of size 3 × 3 to identify the edge area that has been utilized in the embedding process. For each of the three high frequency sub-band (HH, LH and HL), edge blocks are arranged into five groups according to the edge strength. Then, the XOR extraction operations are performed to retrieve n bits from each group as described in Table 3.

5. Experimental results and discussion

There are three standard tests used to evaluate the efficiency of any proposed steganography method. One is the change rate, also known as embedding efficiency, the other is the embedding rate, also known as embedding payload. The third test is the security level which measures the resistance of steganography against steganalysis methods. In this section, we will present the experimental results of our proposed method, which has been implemented in MATLAB R2012b and tested on the BOWS2 database (Bas & Furon, 2007), which contains 10,000 grayscale natural images of size 512 × 512.

5.1. Embedding capacity evaluation

Embedding capacity is an essential measurement to evaluate the performance of steganography methods. It refers to the amount of bits that can be embedded into the cover image. High embedding capacity is an attractive characteristic that most steganography methods strive to achieve. Embedding capacity is computed using Eq. 2.

$$E = \frac{K}{WH}(bpp) \tag{2}$$

where K is the maximum number of secret message bits that can be embedded in the cover image, and W and H are the cover image width and height respectively.

Some steganography methods, such as LSB, provide fixed embedding rate. In our proposed method, embedding payload differs from one image to another, and hence, the embedding rate depends on the contents of the cover image and the threshold value used to discover edges.

5.2. Embedding distortion evaluation

The stego image quality is tested using Peak Signal-to-Noise Ratio ($PSNR$) to evaluate the difference between the cover and stego images, which is calculated using Eq. 3.

$$PSNR = 10\log_{10}\left[\frac{255^2}{MSE}\right](dB) \tag{3}$$

where MSE is the mean square error between cover and stego images, which is defined as:

$$MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(C_{ij} - S_{ij})^2 \tag{4}$$

where C_{ij} and S_{ij} are the gray values of pixel (i, j) of the cover and stego images. W and H are the width and height of the cover image (the stego image has the same size).

$PSNR$ quality measures the distortion occurred on the cover image and it does not take HVS into consideration. The Weighted Peak signal-to-Noise Ratio ($wPSNR$) is an alternate measurement quality, which is defined using Eq. 5. It utilizes an extra parameter called Noise Visibility function (NVF). $wPSNR$ is roughly equivalent to $PSNR$ for flat areas because NVF is close to one in smooth areas. But for edge regions, $wPSNR$ is higher than $PSNR$, because NVF is close to zero for complex regions, and hence it attempts to reflect how the HVS perceives images (Al-Dmour, Ali, & Al-Ani, 2015).

$$wPSNR = 10\log_{10}\left(\frac{\max(x)^2}{\|NVF(C-S)\|^2}\right)(dB) \tag{5}$$

where NVF is defined as (Al-Dmour et al., 2015):

$$NVF_{(i,j)} = \frac{1}{1+\sigma^2_{L_{(i,j)}}} \tag{6}$$

where $\sigma^2_{L_{i,j}}$ denotes the local variance of an image in a window centered on the pixel with coordinates (i, j).

Fig. 8(a) and (b) shows one of the cover image used in the experiment and its histogram. Fig. 9(a)–(c) shows stego images resulting from the 1 bpp embedding algorithm in spatial domain for 5%, 20% and 40% embedding rates. The visual differences between the cover and stego images cannot be discovered by the human eye, and even the histograms of the stego images (illustrated in Fig. 9(d)–(f) are quite similar to that of the cover image.

Fig. 10(a)–(c) show the cover and stego images obtained from 1bbp embedding algorithm for 10% and 30% embedding rates. Smoother and textural parts from both the cover and stego images

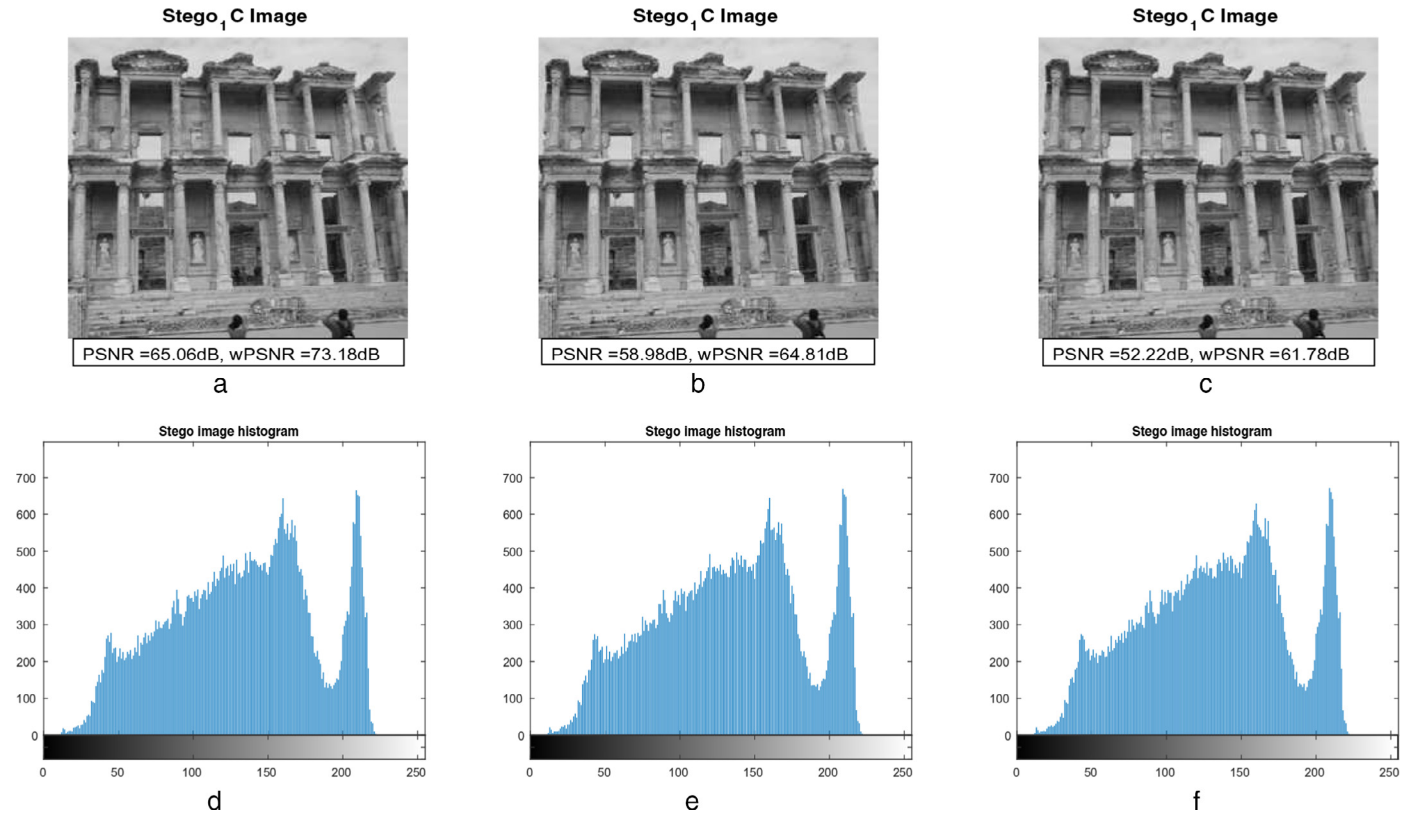


Fig. 9. (a–c) Stego Images using the 1bpp proposed algorithm in the spatial domain with 5%, 20% and 30% embedding rate, and (d–f) Histograms of the corresponding stego images.

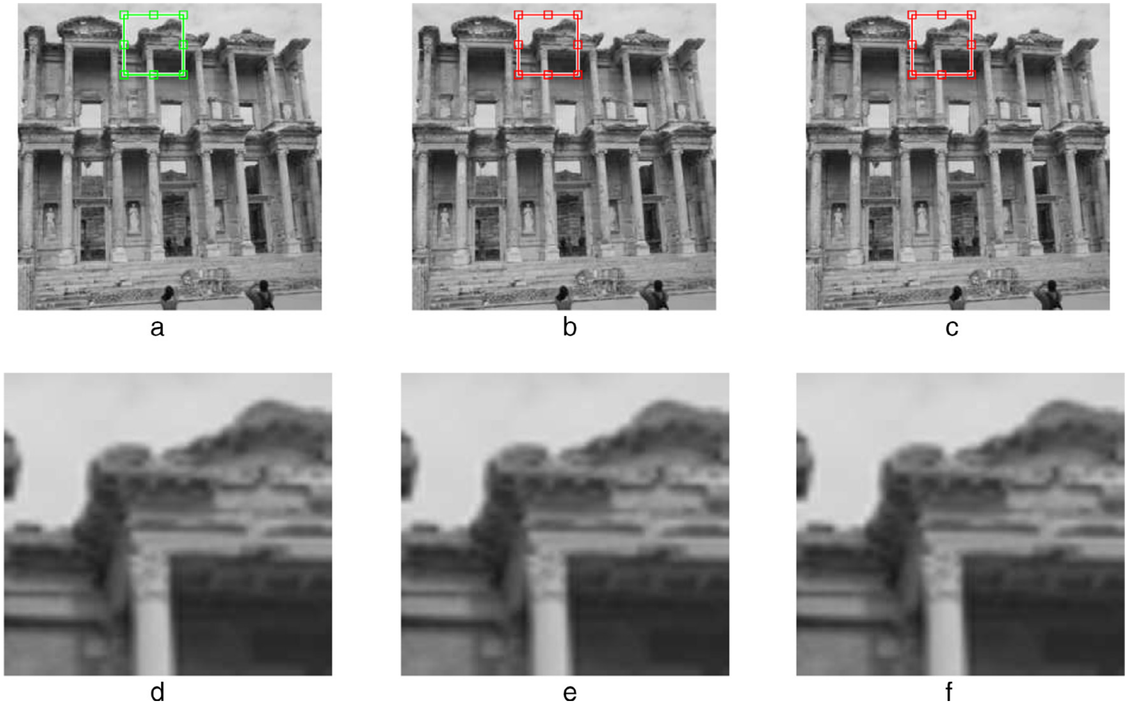


Fig. 10. (a) Cover image, (b–c) Stego Images using the 1bpp proposed algorithm in the spatial domain with 10% and 30% embedding rate, (d) zoomed area from the cover image, and (e–f) zoomed area from the stego image with 10% and 30% embedding rate.

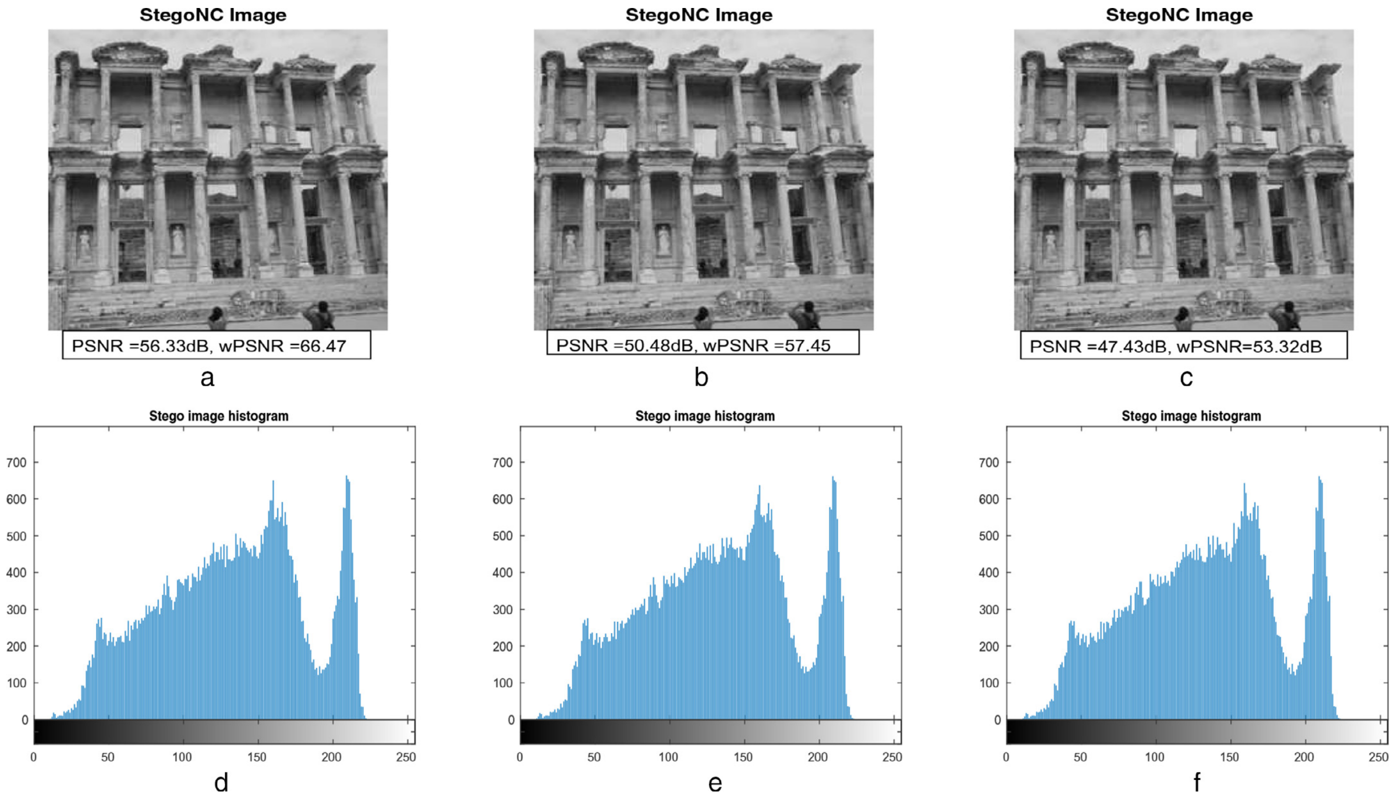


Fig. 11. (a–c) Stego Images using the Nbpp proposed algorithm in the spatial domain with 5%, 20% and 40% embedding rate, and (d–f) Histograms of the corresponding stego images.

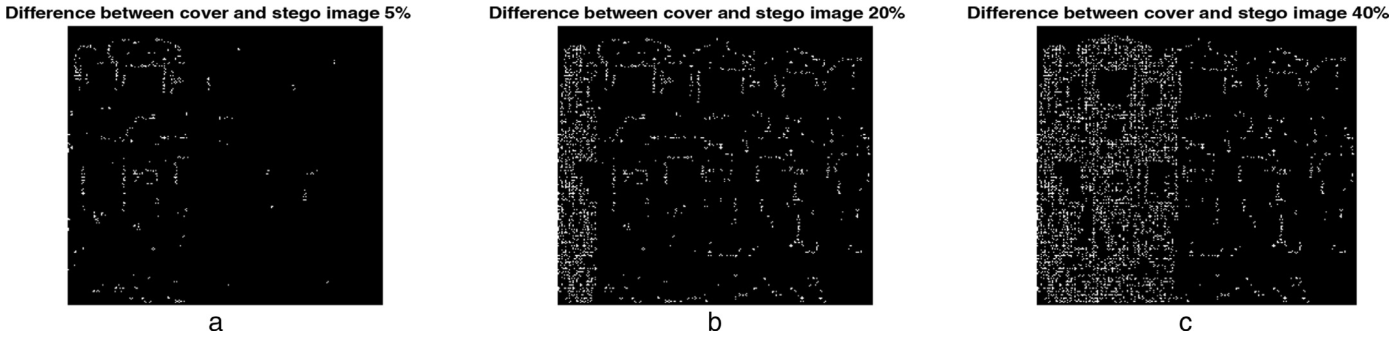

Difference between cover and stego image 5%
a
Difference between cover and stego image 20%
b
Difference between cover and stego image 40%
c

Fig. 12. (a–c) Difference between the cover and stego images using the Nbpp proposed algorithm in the spatial domain with 5%, 20% and 40% embedding rate.

are zoomed as shown in Fig. 10(d)–(f). It can be observed that is hard to notice the difference between the cover and stego images.

Fig. 11(a)–(c) shows the stego images obtained from applying the *n* bpp embedding algorithm for 5%, 20% and 40% embedding rates. The visual difference between the cover and stego images cannot be discovered by the human eye. The stego histogram of the *n* bpp embedding algorithm shown in Fig. 11(d) and (f) indicate a very high degree of similarity with the histogram of the cover image. Fig. 12(a)–(c) represent the difference between the cover and stego images resulting from *N* bpp embedding algorithm with 5%, 20% and 40% embedding rates respectively. The white pixels denote the pixels that have been changed after the embedding process.

Fig. 13(a)–(c) shows the stego images obtained from applying the *n* bpp embedding algorithm in IWT domain for 5%, 20% and 40% embedding rates. The stego histogram of the *n* bpp embedding algorithm shown in Fig. 13(d) and (f) indicate a very high degree of similarity with the histogram of the cover image.

As mentioned in Section 5.2, two measures are commonly used to estimate the quality of the stego images with respect to cover images, which are Peak Signal-to-Noise ration (PSNR) and its weighted version wPSNR. Despite of being widely used, the PSNR quality metric does not take into consideration the HVS characteristics. It calculates the degradation in all regions in the same way. wPSNR has been introduced to give a more accurate image quality measurement. Table 4 presents the quality of the stego images using different 1-bpp steganography methods with embedding rates ranging from 5% to 40%. It is observed that, the proposed 1 bpp method obtained the best image quality compared to EALSB-MR (Luo et al., 2010) since it utilizes the XOR operation to reduce the difference between the cover and stego images. Also, EALSB-MR performs readjustment operation in some cases to guarantee the extraction of the exact secret message in the receiver side. The PSNR values of the TBPC (Hou et al., 2011) and our 1 bpp proposed method are found to be close because both methods use coding where the PSNR values of the TBPC (Hou et al., 2011) and our 1 bpp proposed method are found to be close because both methods use coding to reduce the difference between the cover and stego images, where the TBPC uses a tree structure and our proposed method uses XOR operations. However,

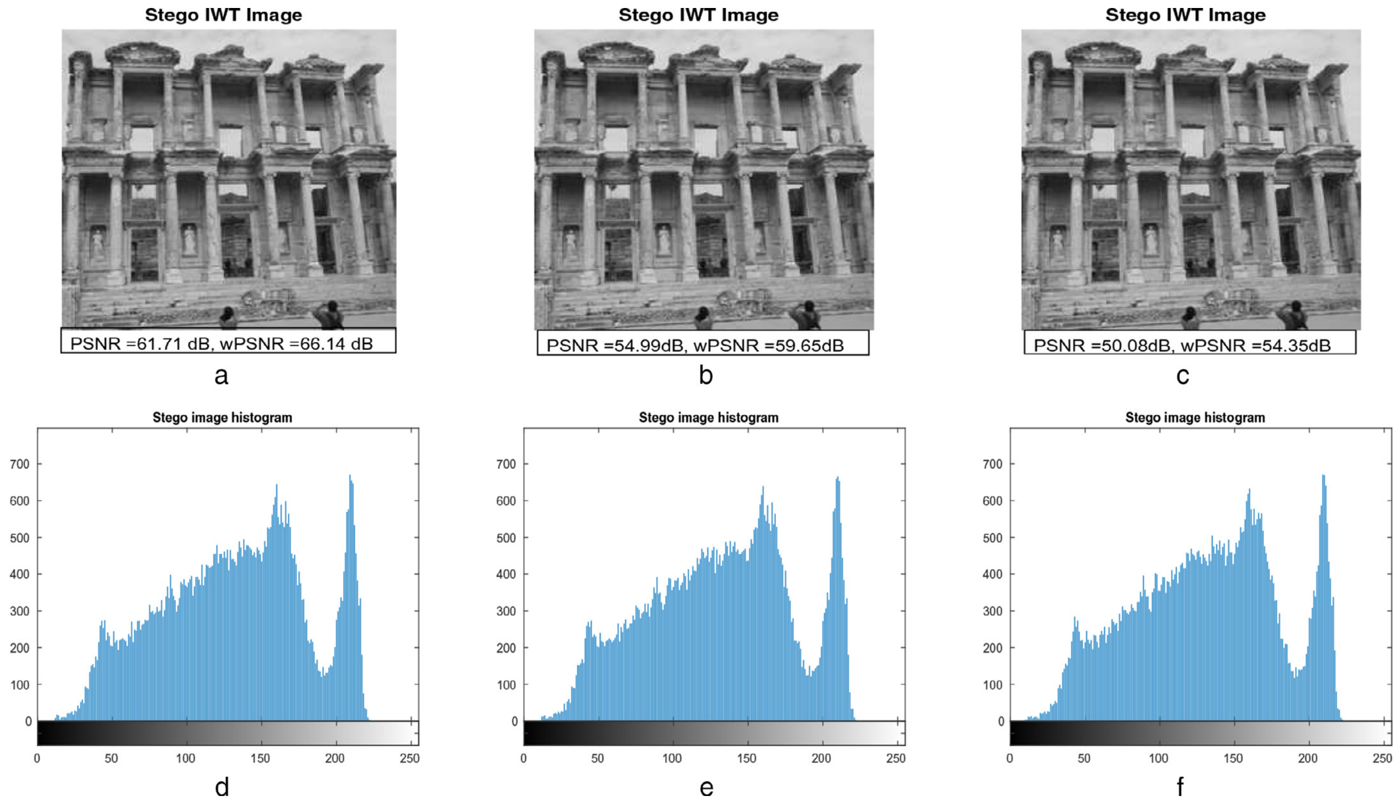


Fig. 13. (a–c) Stego Images using the Nbpp proposed algorithm in the integer wavelet domain with 5%, 20% and 40% embedding rate, and (d–f) Histograms of the corresponding stego images.

Table 4
Image quality evaluation with various 1-bpp steganographics methods in the spatial domain and embedding rates over 10,000 stego images. The red values indicate the best result.

Embedding rate (%)		Method	MSE	PSNR	wPSNR	Avg. difference
	LSB-based	TBPC (Hou et al., 2011)	0.0269	63.82	64.84	0.0269
5	Edge-LSB	EALSB-MR (Luo et al., 2010)	0.0322	63.07	69.99	0.0301
		Proposed	**0.0207**	**64.60**	**71.01**	**0.0207**
	LSB-based	TBPC (Hou et al., 2011)	0.0415	61.94	62.98	0.0415
10	Edge-LSB	EALSB-MR (Luo et al., 2010)	0.0578	60.55	66.23	0.0531
		Proposed	**0.0413**	**61.98**	**66.83**	**0.0413**
	LSB-based	TBPC (Hou et al., 2011)	**0.0809**	**59.04**	60.32	**0.0809**
20	Edge-LSB	EALSB-MR (Luo et al., 2010)	0.1088	57.85	62.14	0.0969
		Proposed	0.0826	58.96	**62.92**	0.0826
	LSB-based	TBPC (Hou et al., 2011)	**0.1012**	**58.99**	60.32	**0.1012**
25	Edge-LSB	EALSB-MR (Luo et al., 2010)	0.1369	56.87	60.69	0.1198
		Proposed	0.1033	57.99	**61.77**	0.1033
	LSB-based	TBPC (Hou et al., 2011)	**0.1230**	**57.23**	58.77	**0.1230**
30	Edge-LSB	EALSB-MR (Luo et al., 2010)	0.1790	55.73	59.02	0.1529
		Proposed	0.1239	57.19	**60.98**	0.1239
	LSB-based	TBPC (Hou et al., 2011)	0.1652	55.95	57.85	0.1652
40	Edge-LSB	EALSB-MR (Luo et al., 2010)	0.2448	54.38	57.20	0.2022
		Proposed	**0.1624**	**56.12**	**60.59**	**0.1652**

the wPSNR values of our proposed method are better than that of the TBPC. The reason is that our proposed method embeds the secret data on the edge regions. Also, the computational cost of the TBPC method is high because of the use of tree structure.

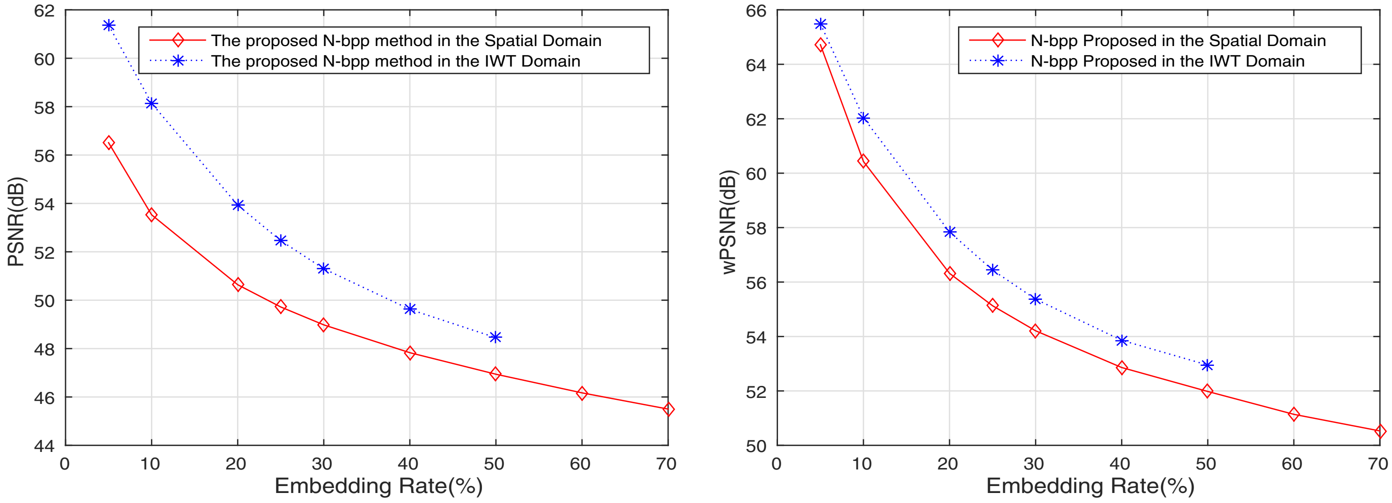
On the other hand, Table 5 presents the results of embedding *n*-bpp using TPVD, edge adaptive PVD, edge adaptive *n*-LSBs and our proposed method. In edge adaptive PVD, the secret data is embedded in the edge regions according to the difference value between each two adjacent pixels. In edge adaptive *n*-LSBs method we expand LSB steganography method and used our edge detection method to discover the sharpest regions for the embedding process. It is similar to our embedding method excluding the XOR operations. The visual quality results are noticeably high for our proposed method compared to the other methods.

The embedding rates listed in Table 5 indicate that the embedding payload of the *n* bpp algorithm is about double that of the 1 bpp algorithm and it exceeds 70% of the cover image size. This is achieved with minor reduction in image quality, as indicated by the PSNR and wPSNR values (shown in Table 5).

Table 6 presents the visual quality performance results of the proposed method in the Integer Wavelet Domain. The results show that the proposed method obtain a good PSNR and wPSNR values for different embedding capacity. The PSNR values are between 61.37 dB and 48.45 dB with 5–50% embedding rates, where the minimum

Table 5
Image quality evaluation with various N-bpp steganographics methods in the spatial domain and embedding rates over 10,000 stego images. The red values indicate the best result.

Embedding rate (%)	Method		MSE	PSNR	wPSNR	Avg. difference
10	Edge-based	Adaptive PVD (Mandal & Das, 2012)	0.675	49.84	58.50	0.1401
		TPVD (Lee et al., 2012)	0.998	48.14	52.11	0.1598
		Adaptive N-LSB	0.307	53.26	60.69	0.0842
	Edge-XOR-based	Proposed	**0.289**	**53.53**	**60.43**	**0.0849**
25	Edge-based	Adaptive PVD (Mandal & Das, 2012)	0.957	48.32	54.74	0.4669
		TPVD (Lee et al., 2012)	2.845	43.59	47.64	0.5154
		Adaptive N-LSB	0.853	48.82	54.73	0.2216
	Edge-XOR-based	Proposed	**0.694**	**49.72**	**55.13**	**0.2085**
40	Edge-based	Adaptive PVD (Mandal & Das, 2012)	1.167	47.46	52.60	0.3784
		TPVD (Lee et al., 2012)	5.742	40.54	45.14	0.6853
		Adaptive N-LSB	1.334	46.88	52.37	0.3446
	Edge-XOR-based	Proposed	**1.072**	**47.83**	**52.86**	**0.3282**
50	Edge-based	Adaptive PVD (Mandal & Das, 2012)	**1.292**	**47.02**	51.59	0.4492
		TPVD (Lee et al., 2012)	6.654	39.90	43.85	0.8914
		Adaptive N-LSB	1.675	45.89	51.35	0.4292
	Edge-XOR-based	Proposed	1.316	46.94	**51.98**	**0.4049**
60	Edge-based	Adaptive PVD (Mandal & Das, 2012)	1.706	45.81	50.58	0.5557
		TPVD (Lee et al., 2012)	11.833	37.40	42.78	1.1052
		Adaptive N-LSB	2.014	45.09	50.43	0.5173
	Edge-XOR-based	Proposed	**1.571**	**46.17**	**51.14**	**0.4842**
70	Edge-based	Adaptive PVD (Mandal & Das, 2012)	**1.816**	**45.54**	50.04	0.6178
		TPVD (Lee et al., 2012)	17.381	35.73	41.70	1.3555
		Adaptive N-LSB	2.350	44.42	49.69	0.6027
	Edge-XOR-based	Proposed	1.833	45.50	**50.53**	**0.5651**



(a) (b)

Fig. 14. (a) PSNR values and (b) wPSNR values of the proposed N-bpp in the spatial and wavelet domains.

Table 6
Image quality evaluation of the N-bpp IWT proposed method with embedding rates over 10,000 stego images.

Embedding rate (%)	MSE	PSNR	wPSNR	Avg. difference
5	0.047	61.37	65.49	0.041
10	0.103	58.14	62.04	0.081
25	0.369	52.46	56.44	0.225
30	0.487	51.30	55.37	0.274
40	0.710	49.62	53.86	0.372
50	0.929	48.45	52.95	0.461

acceptable value of the PSNR is 35dB. Moreover, the maximum average difference between the cover and stego images is up to 0.461.

In order to have a comparison between the two proposed N-bpp methods (spatial and IWT domains), a graphical representation of the PSNR and wPSNR values are shown in Fig. 14(a) and (b). It is clear that the stego images of the proposed IWT method achieves a higher visual quality compared to those obtained using the spatial domain method. However, the spatial domain method provides a higher embedding capacity compared to the IWT method, i.e., 70% compared to a maximum of 50% based on Eq. 2.

5.3. Security evaluation

Security is an important issue in steganography systems. The goal of steganography is to prevent statistical detection by reducing the distortion occurred during the embedding process. In this section, the proposed method is evaluated under blind steganalysis method. Li-110D (Li, Huang, & Shi, 2008) is one of the most efficient steganalyzers used for detecting spatial domain steganography.

Li-110D (Li et al., 2008) extracts statistical moment features of probability density function (PDF) from the normalized histogram of the Local Linear Transform (LLT) coefficients of the image. These

Table 7
The average accuracy value (for 10,000 cover images and their corresponding stego images) against Li-110D with various 1-bpp methods.

Embedding rate	*5%*	*10%*	*20%*	*25%*	*30%*	*40%*	*Average*
TBPC (Hou et al., 2011)	65.51	74.87	81.10	84.06	85.26	89.09	79.98
EALSB-MR (Luo et al., 2010)	49.74	52.14	57.02	59.40	63.21	69.87	58.56
1bpp Proposed	**49.58***	**51.27***	**54.37***	**55.69***	**57.45***	**60.23***	**54.78***

Table 8
The average accuracy value (for 10,000 cover images and their corresponding stego images) against Li-110D with various N-bpp methods.

Embedding rate	*10%*	*25%*	*40%*	*50%*	*60%*	*70%*	*Average*
Adaptive-PVD (Mandal & Das, 2012)	51.89	61.56	74.28	80.44	84.48	86.69	73.22
TPVD (Lee et al., 2012)	68.35	84.90	90.50	92.24	93.55	94.58	87.35
Adaptive-N LSB	54.73	59.02	**62.35***	67.66	69.50	76.37	64.93
N-bpp proposed	**52.80***	**58.83***	62.74	**63.27***	**64.72***	**66.01***	**61.39***

features aim to detect particular alternations of the local texture before and after the embedding process based on the fact that steganography introduces more stochastic textures to the stego images in a fine scale.

Analysis of the proposed method is performed by extracting feature sets from the original and stego images. These features are used to train an SVM (Support Vector Machine) classifier to learn the difference in features produced by data embedding. In our experiments, we selected the linear function kernel with the default parameter setting and used the Matlab implementation of SVM. Also, Testing is done in two-fold cross-validation, i.e., half of the cover and stego images are randomly selected for training, and the remaining images are used for testing. This test is repeated twenty times, and the average of the obtained accuracy values (shown in Tables 7 and 8) indicate that TBPC, EALSB-MR, TPVD, edge adaptive PVD and edge adaptive n LSBs are detected with an accuracy greater than our proposed method for most of the embedding rates (since the number of stego and cover images are equal, the random guess accuracy is 50%). Moreover, when the embedding rate increased, the accuracy value of the SVM classification increased. For example, when the embedding rate is 40% using 1 bpp method, the accuracy value is 60.23%. Even though the obtained classification accuracy values of our proposed method are higher than that of the random guess, these values are not high enough to enable the differentiation between the cover and stego images with an acceptable precision. This however is not the case for other steganography methods, especially with the high embedding rates. Please note that the IWT steganography algorithm has not been tested using this steganalysis method, as the Li-110D has been mainly developed to detect spatial steganography.

6. Conclusion

We presented in this paper an efficient steganography method that makes use of the fact that the human visual system is less sensitive to changes in high contrast areas of the image and therefore attempts to embed the secret message into edge pixels.

The main contribution of the proposed method is introducing new and efficient edge detection algorithm using non-overlapping blocks that estimates the same edge intensities for the cover and stego images. Also, the incorporation of coding theory makes the embedding more efficient. The proposed method that has been implemented in the spatial and wavelet transform domains achieved a good balance between the three steganography evaluation measures of embedding rate, imperceptibility and security when tested on a large dataset that consists of 10,000 images. To be more precise, the stego images produced using the proposed integer wavelet transform algorithm achieved a high level of robustness and visual quality with acceptable embedding rate of upto 50%. On the other hand, the spatial domain algorithm achieved a good embedding payload and maintained low perceptibility, where it achieved a PSNR value of 50.53dB with 70% embedding rate. The proposed method demonstrated considerable improvements in term of image quality and security (tested using a well-established textural steganalysis method) compared with other popular steganography methods. This algorithm is tested on gray images, but it can be extended to color images.

Despite the high-performance results that are achieved in terms of embedding capacity, imperceptibility and security, the proposed method exhibits some limitations. Firstly, the choice between the spatial domain and transform domain implementations is left to the user. Results indicate that the transform domain implementation slightly outperformed its spatial domain counterpart, however, the drawbacks of the transform domain implementation include: (i) relatively lower embedding capacity compared to the spatial domain, and (ii) higher computational cost, as it requires transformation to the wavelet domain. Secondly, the threshold that is used in the embedding and extraction procedure is not automated, and hence, adds to the computational cost of the algorithm.

In our future work, we will attempt to find an alternative way to identify the sharp regions using overlapping blocks in order to enhance the embedding rate. Moreover, we will consider developing an adaptive threshold approach. Finally, in addition to the textural feature steganalysis, we will consider evaluating the security of the proposed algorithm using other steganalysis approaches.

References

Al-Dmour, H., Ali, N., & Al-Ani, A. (2015). An efficient hybrid steganography method based on edge adaptive and tree based parity check. In *Multimedia modeling (mmm 2015)* (pp. 1–12). Springer-Verlag.

Baby, D., Thomas, J., Augustine, G., George, E., & Michael, N. R. (2015). A novel DWT based image securing method using steganography. *Procedia Computer Science, 46*, 612–618.

Bandyopadhyay, D., Dasgupta, K., Mandal, J., & Dutta, P. (2014). A novel secure image steganography method based on chaos theory in spatial domain. *International Journal of Security, Privacy and Trust Management (IJSPTM)., 3*(1), 11–22.

Bas, P. & Furon, T. (2007). Bows-2. http://bows2.ec-lille.fr/.

Bassil, Y. (2012). Article: image steganography based on a parameterized canny edge detection algorithm. *International Journal of Computer Applications., 60*(4), 35–40.

Chan, C.-S., & Chang, C.-Y. (2010). Hiding data in parity check bit. In *Proceedings of the 4th international conference on uniquitous information management and communication 2010 (icuimc 2010).* (pp. 359–363). ACM.

Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: survey and analysis of current methods. *Signal Processing, 90*(3), 727–752.

Chen, W.-J., Chang, C.-C., & Le, T. (2010). High payload steganography mechanism using hybrid edge detector. *Expert Systems with applications., 37*(4), 3292–3301.

Crandall, R. (1998). Some notes on steganography. *Posted on steganography mailing list*.

Geetha, S., Ishwarya, N., & Kamaraj, N. (2010). Audio steganalysis with hausdorff distance higher order statistics using a rule based decision tree paradigm. *Expert Systems with Applications., 37*(12), 7469–7482.

Ghebleh, M., & Kanso, A. (2014). A robust chaotic algorithm for digital image steganography. *Communications in Nonlinear Science and Numerical Simulation, 19*(6), 1898–1907.

Grover, N., & Mohapatra, A. (2013). Digital image authentication model based on edge adaptive steganography. In *2nd international conference on Advanced computing, networking and security (adcons), 2013* (pp. 238–242). IEEE.

Hou, C.-L., Lu, C., Tsai, S.-C., & Tzeng, W.-G. (2011). An optimal data hiding scheme with tree-based parity check. *IEEE Transactions on Image Processing, 20*(3), 880–886.

Hussain, M., & Hussain, M. (2013). A survey of image steganography techniques. *International Journal of advanced Science and Technology. Vol., 54*, 113–123.

Ioannidou, A., Halkidis, S. T., & Stephanides, G. (2012). A novel technique for image steganography based on a high payload method and edge detection. *Expert Systems with Applications., 39*(14), 11517–11524.

Kanan, H. R., & Nazeri, B. (2014). A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm. *Expert Systems with Applications., 41*(14), 6123–6130.

Lee, Y.-P., Lee, J.-C., Chen, W.-K., Chang, K.-C., Su, I.-J., & Chang, C.-P. (2012). High-payload image hiding with quality recovery using tri-way pixel-value differencing. *Information Sciences, 191*, 214–225.

Li, B., He, J., Huang, J., & Shi, Y. Q. (2011). A survey on image steganography and steganalysis. *Journal of Information Hiding and Multimedia Signal Processing. Vol., 2*(2), 142–172.

Li, B., Huang, J., & Shi, Y. Q. (2008). Textural features based universal steganalysis. In *Electronic imaging 2008* (pp. 1201–1212). International Society for Optics and Photonics.

Li, L., Luo, B., Li, Q., & Fang, X. (2009). A color images steganography method by multiple embedding strategy based on sobel operator. In *Multimedia information networking and security, 2009. mines'09. international conference on: 2* (pp. 118–121). IEEE.

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

Mandal, J., & Das, D. (2012). Steganography using adaptive pixel value differencing (APVD) of gray images through exclusion of overflow/underflow. In *Second international conference on computer science, engineering and applications (CCSEA-2012)* (pp. 93–102).

Modi, M. R., Islam, S., & Gupta, P. (2013). Edge based steganography on colored images. In *Intelligent computing theories* (pp. 593–600).

Morkel, T., Eloff, J. H., & Olivier, M. S. (2005). An overview of image steganography. In *Proceedings of the fifth annual information security south africa conference (issa2005), sandton, south africa* (pp. 1–11).

Pal, A. K., & Pramanik, T. (2013). Design of an edge detection based image steganography with high embedding capacity. In *Quality, reliability, security and robustness in heterogeneous networks* (pp. 794–800). Springer-Verlag.

Reddy, H. M., & Raja, K. (2011). Wavelet based non LSB steganography. *International Journal Advanced Networking and Applications. Vol., 3*(3), 1203–1209.

Sharma, P., & Swami, S. (2013). Digital image watermarking using 3 level discrete wavelet transform. In *Proceedings of the conference on advances in communication and control systems-2013 (cac2s 2013)* (pp. 129–133). Atlantis Press.

Thanikaiselvan, V., Arulmozhivarman, P., Chakrabarty, S., Agarwal, A., Subashanthini, S., & Amirtharajan, R. (2014). Comparative analysis of (5/3) and haar IWT based steganography. *Information Technology Journal, 13*(16), 2534–2543.

Verma, N. (2011). Review of steganography techniques. In *Proceedings of the international conference & workshop on emerging trends in technology* (pp. 990–993). ACM.

Westfeld, A. (2001). F5–a steganographic algorithm: high capacity despite better steganalysis. In *Proceedings of fourth international workshop on information hiding, lecture notes in computer science: 2137* (pp. 289–302). Springer-Verlag.

Wu, D.-C., & Tsai, W.-H. (2003). A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters, 24*(9), 1613–1626.

Wu, H.-C., Lee, C.-C., Tsai, C.-S., Chu, Y.-P., & Chen, H.-R. (2009). A high capacity reversible data hiding scheme with edge prediction and difference expansion. *Journal of Systems and Software, 82*(12), 1966–1973.

Zhang, X., & Wang, S. (2004). Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. *Pattern Recognition Letters, 25*(3), 331–339.

Zhu, Z., Zhang, T., & Wan, B. (2013). A special detector for the edge adaptive image steganography based on LSB matching revisited. In *10th ieee international conference on Control and automation (icca), 2013* (pp. 1363–1366). IEEE.

Ziou, D., & Jafari, R. (2014). Efficient steganalysis of images: learning is good for anticipation. *Pattern Analysis and Applications., 17*(2), 279–289.