Multimed Tools Appl (2013) 64:777–793
DOI 10.1007/s11042-011-0975-y

An adaptive LSB matching steganography based on octonary complexity measure

Vajiheh Sabeti · Shadrokh Samavi · Shahram Shirani

Published online: 12 January 2012

Abstract Adaptive steganography methods tend to increase the security against attacks. Most of adaptive methods use LSB flipping (LSB-F) for embedding part of their algorithms. LSB-F is very much vulnerable against simple steganalysis methods but it allows the adaptive algorithms to be extractable at the receiver side. Use of LSB matching (LSB-M) could increase the security but extraction of data at the receiver is difficult or, in occasions, impossible. There are numerous attacks against LSB-M. In this paper we are proposing an adaptive algorithm which, unlike most adaptive methods, uses LSB-M as its embedding method. The proposed method uses a complexity measure based on a local neighborhood analysis for determination of secure locations of an image. Comparable adaptive methods that use LSB-M suffer from possible changes in the complexity of pixels when embedding is performed. The proposed algorithm is such that when a pixel is categorized as complex at the transmitter and is embedded the receiver will identify it as complex too, and data is correctly retrieved. Better performance of the algorithm is shown by obtaining higher PSNR values for the embedded images with respect to comparable adaptive algorithms. The security of the algorithm against numerous attacks is shown to be higher than LSB-M. Also, it is compared with a recent adaptive method and is proved to be advantageous for most embedding rates.

Keywords Steganography · Steganalysis · Adaptive methods · LSB matching · Complexity

1 Introduction

Steganography is an art of sending a secrete message under the camouflage of a carrier content. The goal of steganography is to mask the very presence of communication, making

V. Sabeti · S. Samavi
Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

S. Samavi (✉) · S. Shirani
Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada
e-mail: samavi@mcmaster.ca

the true message not discernible to the observer [5]. On the other hand, steganalysis is the science of revealing the existence of secret messages in a media. The carrier image in steganography is called the “cover image” and the image which has the embedded data is called the “stego image”.

Steganography in LSB is the simplest of methods where data is first compressed and encrypted and then embedded in the LSB of the cover image pixels. LSB embedding can be performed by either LSB flipping (LSB-F) or by LSB matching (LSB-M). In LSB-F technique a data bit replaces the LSB of an image pixel which at most causes a change in the least significant bit of an image. But in the LSB-M method which is first proposed by Sharp, the LSBs are not simply replaced; instead the whole pixel is randomly incremented or decremented if the LSBs differ from data.

Since LSB-M steganography does not create the pair of value (POV) effect it is immune against all attacks that are intended for LSB-F embedding. A number of steganalysis methods are designed for LSB-M and have been successful to some extend but none has been completely effective [16]. In recent years some efforts have been directed toward improving LSB-M which has made it more difficult for the attackers to detect the embedding. In Section 2 of the paper improved versions of LSB-M are discussed and some successful attacks for these methods are reviewed. In LSB-F and LSB-M embedding is the same for all of the pixels of an image. Human vision system (HVS) is sensitive to the changes in the contrasts of an image. Furthermore, HVS is more sensitive toward changes that occur in smooth regions as compared to complex regions with abundance of edges [2]. This means that more data can be embedded in complex regions without creating any suspicion.

A steganographic method using side information is presented in [3] which is called Side Match method. In this method in order to estimate the degree of smoothness or contrast of pixels, correlations among neighboring pixels are calculated. The difference of a pixel with the average of two, three or four of its neighbors is calculated which provides the pixel capacity. The secret message is embedded in the difference value. In [4], Chen *et al.* proposed a modification to the Side Match method. The advantage of the Chen’s method is the increase of the embedding capacity with little degradation in the image quality.

In 2004, Maniccam proposed an information hiding scheme, which begins by taking a pixel to find out its embedding capacity by examining eight of its neighboring pixels. Not all of the pixels of the image are examined in this scheme and LSB-F is used for the embedding part [14].

In [11], Lu *et al.* proposed a method, in which a cover image is quantized to generate extra spaces for hiding secret messages. The cover image is divided into a number of non-overlapped 3×3 blocks. This scheme applies the complexity analysis of neighboring pixels to find out the number of secret message bits that can be embedded in pixels of a block.

The method of Wu *et al.* in [20] is based on Pixel Value Differencing (PVD). They divide the cover image into a number of non-overlapping two-pixel blocks. A large difference between these two pixels indicates that the block is in an edged area and more data can be embedded in it. Hence, PVD is an adaptive method and a number of methods are proposed based on it. In these methods, the secret message is embedded by using of LSB-F and PVD. Sabeti *et al.* successfully attacked some versions of PVD [17, 18].

In adaptive methods images that are more complex and contain more edges have higher embedding capacities as compared to images with more smooth regions. Since the stego image is not identical to the cover image, a big hurdle in adaptive methods is to identify embedded regions in the stego image at the time of the extraction of data. This means that a pixel may be identified as suitable in the embedding process and the embedding may cause it to change. Then, the changed version of it may be identified as non-edge and hence the extraction process fails to recover data from that pixel. To circumvent this problem, the stego image is tried to have as little change in it as possible. To achieve this, adaptive methods usually employ LSB-F as opposed to LSB-M. Only the algorithm proposed in [13, 21] uses LSB-M for embedding while they only use two neighboring pixels to identify a complex region. These identified edges that are based on only two pixels are not very reliable.

Reliable and accurate identification of a complex region can play an important role in an adaptive method. To achieve this accuracy an appropriate neighborhood of the pixel should be looked at. In some adaptive algorithms fixed blocks (such as 3×3) are used but not all of the pixels of an image are analyzed and hence not all of the pixels are candidates for embedding [11, 14]. These methods suffer from lack of appropriate capacity. In some other methods, a combination of the neighboring pixels such as top, bottom, left, or right pixels are used to distinguish a smooth region from a complex one [3, 4].

In this paper an algorithm is presented which intends to alleviate the shortcomings of other methods. Our main goal is to improve the security as compared with comparable adaptive methods. We call this method as Complexity Based LSB-M (CBL) algorithm. Firstly, the proposed CBL method uses an 8-neighborhood of a pixel to identify suitable complex regions of an image for embedding purposes. This means that regional complexities can be analyzed more realistically than methods that only use two adjacent pixels. Secondly, these 3×3 blocks are overlapped so that all of the pixels are studied which increases the capacity of the algorithm compared to comparable adaptive algorithms. Thirdly, unlike many other adaptive algorithms, LSB-M is used for the embedding process, which has higher security than LSB-F. Provisions are put in place to make sure that when a pixel is identified as complex in the embedding process, the extraction process can correctly identify that pixel and hence, the correct data is extracted.

The paper is organized such that in Section 2 a number of LSB-M based algorithms and some successful attacks for them are presented. The proposed algorithm is presented in Section 3. Implementation results of the algorithm and the results from steganalysis of the method are discussed in Section 4. Concluding remarks and suggestions for the continuation of the work are in Section 5 of the paper.

2 Related works

Since the proposed method is based on LSB-M, in this section we go over some steganography methods that try to improve LSB-M.

In the LSB matching, when the value of a pixel is to changed it is randomly either increased or decreased. In [15], pixels are modified such that the outcome of the binary function is equal to the desired data value. The pair of pixels performs as a unit, where the LSB of one pixel carries one bit of information, and the binary function contains the other bit of information. This method allows embedding of the same payload as LSB matching but with fewer changes to the cover image. It shows a better security than LSB matching.

Li *et al.*, in [10], proposed a generalized LSB matching (G-LSB-M) scheme by generalization of the method in [15] and LSB matching. Liu *et al.*, in [12], proposed a content-adaptive scheme which they claimed to have better security as compared with LSB-M and G-LSB-M. In their method, if the secret message bit does not match the LSB of the corresponding cover pixel value, the choice of modification direction is not random and is tried to have the best correlation with the neighboring pixels.

In 2010, Luo *et al.* proposed an algorithm which is called ALSBMR [13]. This is one of the rare adaptive methods which uses LSB-M for the embedding process. Luo *et al.* use a combination of the idea of LSB-M, known as LSB-M revisited [15], and the idea of adaptive embedding. It chooses pairs of pixel to determine the complex regions of the image, suitable for embedding.

To show the security of our algorithm we used five steganalysis attacks which we briefly explain in the followings. One of the detectors for LSB-M in the literature belongs to Harmsen *et al.* [7]. This method relies on the fact that LSB-M tends to smooth out the histogram of an image. This causes a shift of the center of mass of the histogram's spectrum toward the origin. To address this shift of center of mass in the histogram's spectrum, Ker [9] proposed two methods of applying the histogram characteristic function (HCF). These methods are based on (1) calibrating the output through the use of a downsampled image, and (2) obtaining the adjacency histogram instead of the usual intensity histogram. Hence, major improvements in detection of LSB-M in grayscale images became possible. We refer to these two methods as Ker1 and Ker2.

In [6], Goljan *et al.* proposed a blind steganalysis technique which estimates the stego noise through denoising of the detail bands of a first order wavelet decomposition of an image. Hence, this method is referred to as Wavelet Absolute Moment (WAM) steganalysis. They use a feature vector consisting of 27 moments (9 per band).

Zhang *et al.* [22] presented a scheme to work on the local extrema of the histogram. The filtering operation can reduce the amplitude of local extrema. Cancelli *et al.* [1] extended this strategy by analyzing four 2D adjacency histograms In addition to the other mentioned features, these adjacency histograms result in a 10-dimensional feature vector. We refer to this attack as Amplitude of Local Extrema (ALE) method. In [8], Huang *et al.* suggested a method for detection of the LSB-M stegonagraphy when applied to uncompressed gray scale images. An image is formed by combining the least two significant bit-planes and partitioning it into 3×3 overlapped blocks. The blocks are categorized into four types based on their number of gray levels. Through embedding a random sequence by LSB matching and then computing the alteration rate of the number of elements in group 1, it is claimed that normally the alteration rate is larger in the cover image than in the stego image. We refer to this attack as CNGL.

3 Proposed method

In this section we present a general description and definitions used in an adaptive steganography method, which embeds data in spatial domain based on the complexity features of an image. Then based on the presented definitions we explain the proposed CBL steganography method.

Let us assume that an $m \times n$ cover image is described as a set I, consisting of elements $I(i, j)$ such that

$$I = \{I(i,j) | 1 \leq i \leq m, 1 \leq j \leq n\} \tag{1}$$

The secrete data (message) consists of elements $d(k)$ and when embedded in the cover image will generate the stego image S with $S(i, j)$ as its elements as described in Eqs. 2 and 3.

$$Data = \{d(k)|1 \leq k \leq w\} \tag{2}$$

$$S = \{S(i,j)|1 \leq i \leq m, 1 \leq j \leq n\} \tag{3}$$

The cover image could be colored, grayscale, or black and white. It is assumed that the length (number of data elements) of the message is w. In adaptive embedding methods it is possible that some of the pixels are not embedded in and they are only used to determine that capacity of other pixels. In another words, $I = [I', I'']$, where data set I' consists of the used pixels and data set I'' is the set of the reference pixels. Embedding is only performed on the I' pixels and only after the capacity of each member pixel is determined. The capacity depends on the local complexity of that pixel which in turn is a function of the value of that pixel and those of some of its neighboring pixels. Hence, the complexity of a pixel $I(i, j)$ is defined as:

$$complexity(I(i,j)) = f(I(i,j), N(i,j)), \quad \forall I(i,j) \in I' \tag{4}$$

where $N(i, j)$ is a set of some of the neighboring pixels:

$$N(i,j) = \{I(i+u, j+v)|u \subset [-m, m], v \subset [-n, n]\} \tag{5}$$

Then the capacity, $C(I(i, j))$, for a pixel $I(i, j)$ can be a function of the complexity of that pixel.

$$C(I(i,j)) = f'(complexity(I(i,j))) \tag{6}$$

In many of the existing adaptive methods, to determine the capacity of a pixel a set of threshold values are used. When the capacity of a pixel is determined as $c = C(I(i, j))$, then

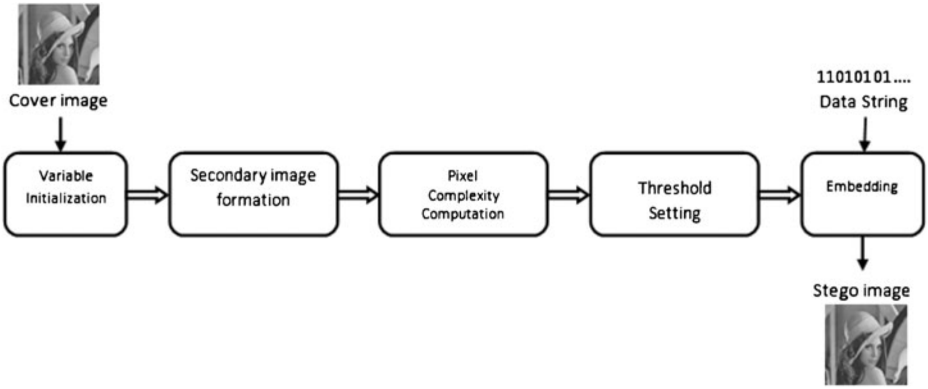

Fig. 1 Steps of embedding procedure

data elements $D = \{d(q), d(q+1), \ldots, d(q+c-1)\}$ are embedded in that pixel. Hence, the embedding function *Emb* (.) applies to each member of I' using the local complexity of each member pixel. The embedding procedure produces stego pixels, *S*(*i*, *j*).

$$S(i,j) = Emb(I(i,j), D) \tag{7}$$

Different adaptive algorithms present different embedding functions. For every embedding function there should exist an extraction function, *Ext*(.), where the initial embedded data is recovered from the stego image:

$$D = Ext(S(i,j)) \tag{8}$$

Now we explain the proposed algorithm. Basic steps of the embedding algorithm are shown in Fig. 1.

In the LSB-M embedding algorithm all of the pixels have the same priority for embedding and the only means of knowing which pixel is embedded in is the key that is shared between the sender and receiver of the image. No means of prioritization of pixels are employed.

To identify smoothness of a region in an image we need to define a criterion. In the followings we refer to *complexity* as a criterion. For embedding based on complexity we need to measure the complexity of each pixel. Embedding is only performed for pixels that posses complexity value which is higher than a certain threshold. An important point is that the receiver side should be able to accurately identify the pixels that are embedded in. Any inaccuracy in the extraction phase could result in the loss of the entire data which is usually compressed and encrypted prior to the embedding process.

In following subsections details of each part of the algorithm and the extraction procedure are explained.

3.1 Variable initialization

It is necessary to initialize a number of variables at the beginning of the process. A pseudo random number generator (PRNG) is to be used at different stages of the algorithm. Hence, a seed is to be selected and shared between the sender and receiver. Let *I* indicate the cover image which has *m* lines and *n* columns. A copy of this image is saved as *O*. Also, let *I*(*i*, *j*) indicate a pixel at row *i* and column *j* of image *I*. The following are performed at the initialization step:

```
set PRNG seed;
$I \leftarrow$ cover image;
$O \leftarrow$ copy of cover image;
$[m, n] \leftarrow$ dimensions of cover image;
```

3.2 Secondary image formation

In this step a secondary image is formed by modifying the initial cover image *I*. This image will be used for computation of the complexity values of pixels. The goal is that the receiver can replicate this process and come up with the same set of complexity values. The secondary image pixels are forced to have zero at their LSBs. This is performed by the following routine.

```
for each pixel $I(i,j)$
        $r \leftarrow$ random number $\in [0,1]$;
        if $I(i,j)$ is odd then
                if $r \leq 0.5$ then
                        $I(i,j) \leftarrow I(i,j) + 1$;
                else
                          $I(i,j) \leftarrow I(i,j) - 1$;
                end if
        end if
end for
```

3.3 Pixel complexity computation

As was mentioned at the beginning of Section 3, some adaptive methods consider an image as $I = [I', I'']$, where only the I' pixels are embedded in. In our proposed algorithm all of the image pixels are potentially candidates for embedding. In another words, in our method $I = [I']$. Hence, for every pixel $I(i, j)$ a variable *complexity* (i, j) is computed. The value of this variable shows the type of region that the pixel belongs to. One can come up with different means of defining this complexity criterion. What we suggest is to use the sum of absolute values of differences of the pixel with its neighbors. Different types of neighborhoods can be defined. But the correlation between the value of a pixel and its neighboring pixels is reduced as the distance between them increases. Hence, the use of 8-neighbor seems to give a good sense of the type of region that the pixel is residing in. We therefore refer to the mentioned complexity measure as octonary. Figure 2 shows the neighborhood for pixel $I\ (i, j)$.

Equation 9 shows how *complexity* (i, j) is computed:

$$complexity(i,j) = \sum_{u=-1}^{1} \sum_{v=-1}^{1} |I(i,j) - I(i+u, j+v)| \tag{9}$$

Based on the above definition of complexity, we see that the higher value of *complexity* (i, j) means that pixel $I(i, j)$ is in a location where large fluctuations in the intensities of pixels are experienced. Similarly, lower values of complexity indicate smooth regions. Hence

$I(i-1, j-1)$	$I(i-1, j)$	$I(i-1, j+1)$
$I(i, j-1)$	$I(i, j)$	$I(i, j+1)$
$I(i+1, j-1)$	$I(i+1, j)$	$\mathrm{I}(i+1, j+1)$

Fig. 2 Pixel $I\ (i, j)$ and its 8-neighborhood

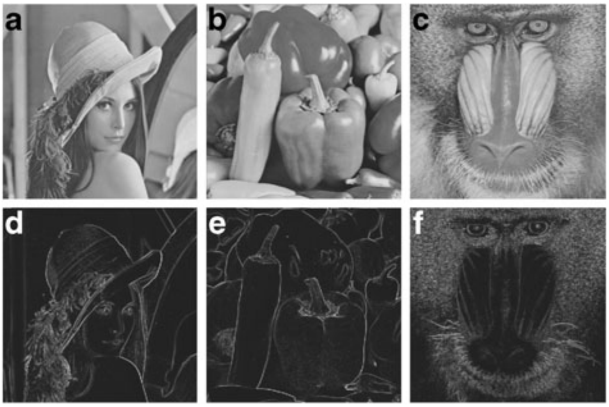

Fig. 3 Test images and their corresponding octonary complexity values of **a**,**d**) Lena, **b**,**e**) Peppers, **c**,**f**) Baboon

larger values of *complexity* (i,j) identify the pixel as appropriate for embedding purpose. For further illustration three example images of Lena, Peppers, and Baboon are shown in Fig. 3. Also shown in this figure are corresponding images of the octonary complexity measures. For illustration purposes complexity values are scaled. We see that smooth regions have smaller complexity values and are illustrated by dark pixels in Fig. 3d, e and f.

3.4 Threshold computation

The complexity value of a pixel alone cannot identify a pixel as belonging to a smooth area or otherwise. Complexity by itself is a relative criterion and requires a threshold value to be compared with. The result of this comparison can then be used to categorize the pixel as belonging to an edge or smooth area.

To come up with a suitable threshold we need to know the fraction of the capacity of the image that is going to be embedded. This is done by knowing the number of bits in the secrete message and the number of pixels of the image. If p percent of the pixels are to be embedded then an appropriate threshold T can be computed. The value of T is chosen such that at least p percent of the pixels are labeled as complex and are embedded with data. Equation 10 shows how T is computed.

$$T = max\{t_0 || \{(i,j)|complexity(i,j) \geq t_0\}| \geq (p * m * n/100)\} \quad 0 \leq p \leq 100 \tag{10}$$

Table 1 Threshold values for different embedding rates and different images

Image name	Percent of embedded pixels (p)		
	30	50	80
Lena	50	32	18
Pepper	52	36	22
Baboon	186	112	42

While the value of T is dependent on p it is mainly an attribute of the image textures and pattern. For a given p different T values may be produced by different images. This is illustrated in Table 1 where three images of Fig. 3 are tested with p values of 30, 50, and 80. We see that threshold values vary in the range of 18 to 182 for these specific images. Lena image has more smooth regions as compared to the Baboon image. That is why threshold values in Baboon is much higher than those of Lena.

3.5 Embedding

After the appropriate pixels are labeled, the embedding process is performed in those pixels. The following shows the embedding procedure, where d is the string of message bits:

```
$set\ PRNG\ seed;$
$k \leftarrow 0;$
$for\ each\ pixel\ I(i,j)$
        $r \leftarrow random\ number \in [0,1];$
        $if\ I(i,j) == 0\ or\ complexity(i,j) < T$
                $S(i,j) \leftarrow O(i,j);$
        $else$
                $if\ I(i,j)\ mod\ 2 \neq d(k)$

                        $if\ r \leq 0.5\ then$
                                $S(i,j) \leftarrow I(i,j) - 1;$
                        $else$
                                $S(i,j) \leftarrow I(i,j) + 1;$
                        $end\ if$
                $end\ if$
                $k \leftarrow k + 1;$
        $end\ if$
$end\ for$
```

It should be noted that the same *PRNG* is used with the same seed as was used in the I image formation. This procedure guarantees that the value of each pixel, which has a complexity value lower than T, is replaced by its corresponding value in the O image. This means that those pixels that are not embedded in are not changed at all. This reduces any unnecessary changes in the image.

The value of those pixels that have complexity values larger than T must be changed such that their LSB values correspond to the message bits. Since the LSB of all pixels of image I were changed to zero, if the intended embedding bit is zero there is no need to change that pixel. But if the message bit is 1 then the pixel value has to be either increased or decreased. If a pixel originally had an LSB of 1 is being embedded with a 1 then the original pixel value is produced in this embedding procedure.

3.6 Extraction procedure

Extraction procedure at the receiver should be performed so that the exact embedded data string is obtained. Based on basic ideas of steganography, the receiver has no access to the cover image. Hence, the extraction procedure, just by knowing the T value, should be able to correctly identify the complex pixels and obtain the message. Once the complex pixels are identified the extraction process is the same as the LSB matching algorithm. This means that the LSB of the complex pixels will comprise the embedded data.

The receiver should implement the same four steps that are performed in the transmitter to correctly identify the embedded pixels. The only difference is that the transmitter performed these four steps on the cover image and the receiver implements these steps on the stego image. The required steps for the extraction process are shown in Fig. 4.

The following pseudo code illustrates the detail of the extraction procedure.

```
$k \leftarrow 0;$
$for\ each\ pixel\ S(i,j)$
    $if\ S(i,j) \neq 0\ and\ complexity(i,j) \geq T$
        $d(k) \leftarrow mod(S(i,j), 2);$
        $k \leftarrow k + 1;$
    $end\ if$
$end\ for$
```

4 Implementation results

Since CBL is an adaptive algorithm it is expected that most of data is embedded in edges. To illustrate this phenomenon, we can produce the difference between the cover and the stego images. In Fig. 5 the difference images for three embedding rates of 0.2 bpp, 0.3 bpp and 0.5 bpp are shown for Lena cover image. White spots in difference images are pixels where the embedding was performed. As the embedding rate is increased more edges are pronounced in the difference image.

4.1 CBL versus adaptive LSB-F based methods

For an adaptive algorithm such as the suggested CBL, it seems that it should be compared with some of the existing adaptive algorithms. But there is no specific steganalysis method for these adaptive methods. Hence, we use the stego image quality based on PSNR as the means of comparing our algorithm with some of the adaptive algorithms. Three 512×512-pixel standard

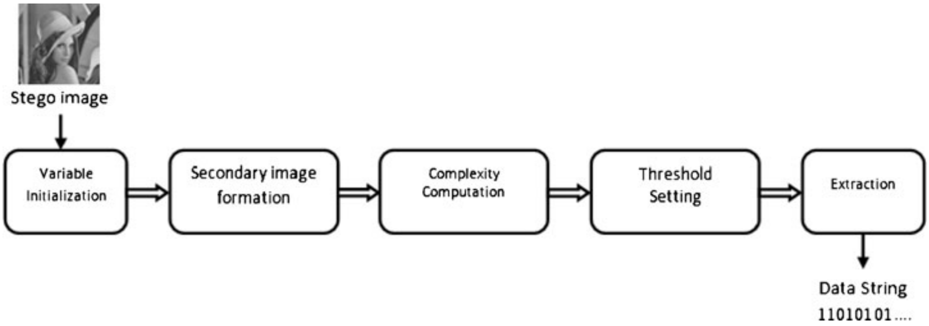

Fig. 4 Steps for extraction procedure

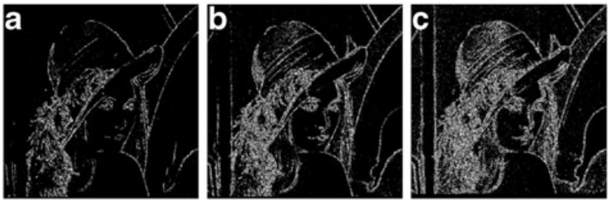

Fig. 5 Difference images with embedding rate of **a**) 0.2 bpp, **b**) 0.3 bpp, **c**) 0.4 bpp

images of Lena, Baboon, and Peppers, as shown in Fig. 3, are used to compare CBL with other algorithms in terms of PSNR.

Three other adaptive algorithms that are compared with our CBL are Maniccam [14], Side Match [3], and PVD [20]. Each algorithm is used to embed three different data rates of 0.3 bpp, 0.5 bpp, and 0.8 bpp. Table 1 compares PSNR values of the stego image as compared with the cover image for the four comparing algorithms. Since Maniccam was not able to embed at 0.5 and 0.8 bpp we left the corresponding PSNR values in Table 2 as blank. It is noticed that our CBL algorithm in all of the images and all of the embedding rates would produce better PSNR values. This advantage of CBL is due to the fact that we only change a pixel by one grey level while other algorithms may embed more than one bit in a pixel.

4.2 CBL versus non adaptive LSB-M based methods

From another point of view we need to compare CBL with other LSB-M based methods. A recent LSB-M based method is CAS-NE [12] which is more secure than other comparable algorithms such as G-LSB-M [10]. Any of the steganalysis methods mentioned in Section 2 can be used for comparison purposes. We used Ker1, Ker2, WAM, ALE, and CNGL attacks to LSB-M, our CBL, and CAS-NE algorithms. Out of the mentioned five attacks, WAM is a blind method and the four are specifically designed to discover LSB-M embedding.

Different image databases exist that are used for testing of steganography algorithms. Images of each database usually have certain characteristics. Hence we used two different databases. The first one was *NRCS Photo Gallery* which is maintained by

Table 2 Comparison of PSNR values between CBL and other adaptive algorithms

Method	Percent embedding	Lena	Peppers	Baboon
Maniccam	0.3 bpp	43.10	43.06	43.17
	0.5 bpp	×××	×××	×××
	0.8 bpp	×××	×××	×××
Side Match	0.3 bpp	47.29	43.92	36.99
	0.5 bpp	44.18	42.03	34.97
	0.8 bpp	41.23	40.13	32.95
PVD	0.3 bpp	51.77	49.77	41.98
	0.5 bpp	48.70	47.22	40.35
	0.8 bpp	45.57	44.70	39.39
CBL	0.3 bpp	56.39	56.37	56.36
	0.5 bpp	54.16	54.14	54.13
	0.8 bpp	52.11	52.12	52.11

the United States Department of Agriculture [19]. This database has 2,375 photos related to natural resources such landscapes from across the USA. The second database that we used was *Camera Images* that is a collection of 3,164 images captured using different digital cameras by researchers from Binghamton University, NY, USA. Images of natural landscapes, buildings and object details are included in this database without applying any lossy compression.

Implementation of different algorithms and different attacks were performed by Matlab 7.6.0. Results of a number of experiments on images using different embedding rates are presented in the following. Receiver operating characteristic (ROC) curves were used to show the results of the experiments.

Figure 6 shows the results from attacks of Ker1, Ker2, CNGL and ALE attacks when database images where 50% embedded. We have also experimented with 25% and 80% embedding rates but their ROC curves are not presented here. It is concluded that the securities of LSB-M and CAS-NE against Ker1 attack are the same and are degraded as the embedding rate is increased. On the other hand the security of our CBL against Ker1 attack is much better and even at 80% embedding it remains undetected.

Results from Ker2 attack are shown in Fig. 6b. These results too are from embedding with 50% rates in images of NRCS database. While at 25% embedding rate CBL has not much advantage over LSB-M or CAS-NE, when embedding rate increases the higher security of CBL becomes apparent.

Another attack which was applied was CNGL [8]. This attack is designed to detect LSB-M embedding in uncompressed images. Hence we applied our CBL, LSB-M, and CAS-NE method to embed data in images of Camera database which contains uncompressed images. Different embedding rates were again used to compare these three algorithms. Figure 6c shows the ROC curve produced from the application of the CNGL attack. The superiority of our CBL is apparent even at low data rates. This particular steganalysis operates on changes that occur in smooth regions of an image. Since CBL does not embed in smooth regions, it

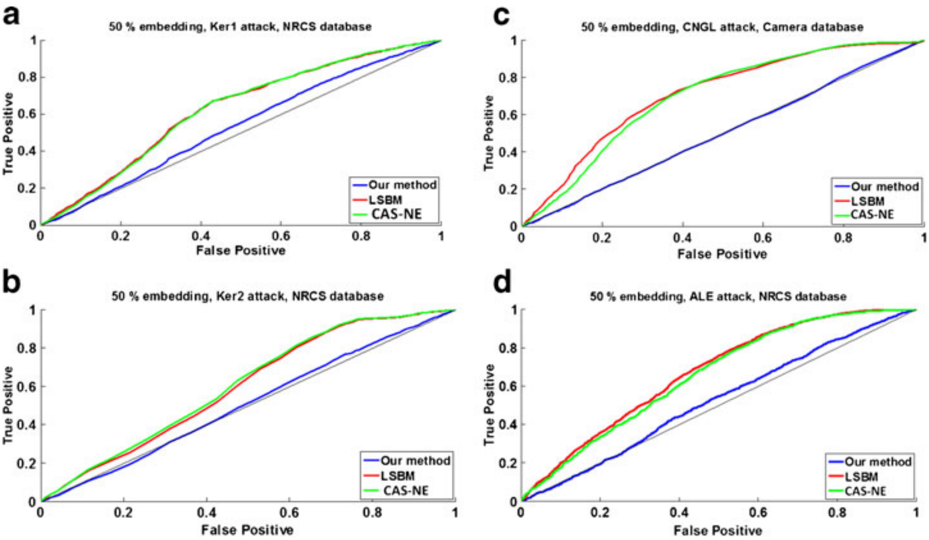

Fig. 6 Security of proposed CBL compared with LSB-M and CAS-NE for embedding rate of 50% when attacked by **a**) Ker1, **b**) Ker2, **c**) CNGL, and **d**) ALE

has very low vulnerability to this attack. Even though CAS-NE is more successful than LSB-M but it has no defence against CNGL for uncompressed images.

Steganalysis attack of ALE [1] was also tested to compare our CBL algorithm with the other two LSB-M schemes. An ROC curve representing the comparison of the security of algorithms against ALE are shown in Fig. 6d. CBL turns out to be completely secure against ALE as opposed to the other two schemes which at higher embedding rates become less secure.

So far, the attacks that were presented were all specifically designed to detect LSB-M. There are blind attacks that are designed to detect any type of embedding algorithm. A blind attack that is used in many of the references to detect LSB-M and similar algorithms is WAM. Hence, we used this last attack to test the security of CBL against the security of the other two schemes for embedded images of Camera database. Figure 7a shows comparisons of the security of the three embedding algorithms against WAM. This attack is more successful in detecting LSB-M based algorithms but our CBL has lower vulnerability against WAM in comparison with the other two schemes. At all three embedding rates our ROC curves fall below those of LSB-M and CAS-NE.

We also applied WAM to images of NRCS database, embedded with the mentioned three schemes. Figure 7b shows the results for this attack. The WAM attack on NRCS images is less successful as compared with the results obtained from the Camera database. Our experiments show that for embedding rates below 70% our CBL performs better than CAS-NE and LSB-M. Only in embedding rates that are higher than 70% we see that CBL's performance falls below that of CAS-NE. This is due to the fact that when embedding rate increases towards 100% and all of the pixels of an image are to be embedded, then CBL has no advantage over LSB-M. The strength of CBL comes from its selection of more secure regions. Hence, at lower embedding rates superiority of CBL is more apparent.

4.3 CBL versus ALSBMR

It is mentioned in [13] that their adaptive steganography algorithm, called ALSBMR, is the most secure LSB-M based method. Therefore, we compare our work with ALSBMR for a number of targeted attacks. The comparison was performed on some embedded images using all of the 5 previously mentioned attacks. Table 3 presents the results of these attacks against the proposed CBL, ALSBMR, and LSB-M. Security of each algorithm was measured by plotting ROC curves and extracting two different features from each curve.

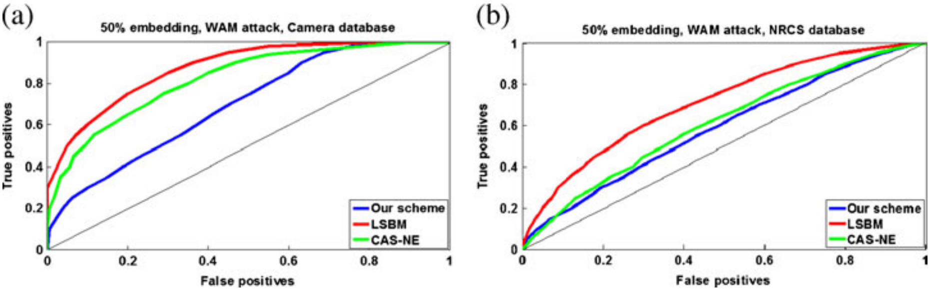

Fig. 7 Security of proposed CBL compared with LSB-M and CAS-NE for embedding rate of 50% when attacked by WAM for image database of **a**) Camera and **b**) NRCS

One measure of security of an algorithm, indicated by $\{FA\}_{50\%}$, is the rate of false alarms that are produced at the 50% true positive point on a ROC curve. The closer that $\{FA\}_{50\%}$is to 50% point the more secure is the steganography method and the performance of the attack is closer to random guessing. Another measure of security is *Accuracy* which is the area between the ROC curve and the diagonal. This area is normalized so that a perfect detection has an accuracy of 1. When this area is zero the ROC curve is the same as diagonal and the attack's outcome is identical to random guessing. In Table 3, bolded values indicate occasions that our method is more secure than the other methods.

It can be seen from Table 2 that the security of CBL is either higher or comparable with ALSBMR. Besides having better security, CBL has other advantages of:

1- CBL can be used in any types of images, while ALSBMR could fail in some cases. There is a readjustment stage in ALSBMR which requires an extracted threshold from the image, for the required amount of embedding, to be less than 31. For situations that the image has plenty of edges and the amount of embedding is low, the readjustment stage fails. CBL has no readjustment stage and always successfully embeds in any type of image.
2- In ALSBMR only the threshold value is transmitted to the receiver. There is a possibility that the embeddable pixels for the transmitted threshold are more than the number of data bits. This causes an ambiguity at the receiver side. In CBL the length of data stream is specified and hence certainty exists during extraction stage.
3- Both ALSBMR and CBL use LSB-M for embedding, hence, both methods embed, at most, one bit per pixel. While in this paper we only used maximum of one bit per pixel, CBL has the advantage of generality and can embed more bits in each pixel.

Table 3 Security comparison between CBL, LSB-M and ALSBMR against 5 different attacks

Attack	Embedding method	Percent of embedded pixels (p)					
		25%		50%		70%	
		Accuracy	$\{FA\}_{50\%}$	Accuracy	$\{FA\}_{50\%}$	Accuracy	$\{FA\}_{50\%}$
Ker1	CBL	0.0503	0.4721	0.2303	0.2989	0.3176	0.1895
	ALSBMR	0.0158	0.5	0.0169	0.5065	0.1056	0.4120
	LSBM	0.0063	0.4486	0.1539	0.3631	0.3299	0.2835
Ker2	CBL	0.0120	0.4944	0.0713	0.4727	0.1085	0.4714
	ALSBMR	0.0092	0.5055	0.0132	0.5129	0.1575	0.3486
	LSBM	0.1134	0.3665	0.4376	0.1749	0.7243	0.0751
CNGL	CBL	0.0509	0.5754	0.0488	0.4588	0.0536	0.4497
	ALSBMR	0.0854	0.4603	0.0860	0.4375	0.2101	0.3251
	LSBM	0.4651	0.2226	0.6992	0.1508	0.7842	0.1061
ALE	CBL	0.0398	0.4749	0.2601	0.3296	0.5124	0.1955
	ALSBMR	0.0553	0.4737	0.2807	0.3184	0.5329	0.1788
	LSBM	0.3869	0.2570	0.6618	0.0838	0.7694	0.0503
WAM	CBL	0.2845	0.3324	0.5881	0.1844	0.8026	0.0355
	ALSBMR	0.4414	0.2222	0.7096	0.0838	0.8865	0.0112
	LSBM	0.7743	0.0503	0.9054	0.0168	0.9360	0.0056

5 Conclusion

In adaptive algorithms the embedding rate for each pixel of an image is determined by the human visual system. In these algorithms higher embedding rates are performed in the edge areas of images. Smooth regions are either left alone or lower embedding rates are performed in them. A short coming of the existing adaptive algorithms is that they embed data using LSB-F routine. We eased this problem by embedding based on LSB-M algorithm. Furthermore, we presented a complete scheme which identifies edges in the cover image and after altering the image by embedding data in it, allows the receiver to identify the exact same edges for the extraction purposes.

Two large databases were used to show the performance of our CBL algorithm as compared to three comparable algorithms. Both blind and specific steganalysis methods were applied to the produced stego images. These experiments showed the superiority of our algorithm over LSB-M, CAS-NE (non adaptive and LSB-M based) and ALSBMR (adaptive and LSB-M based) algorithms.

The proposed CBL algorithm is similar to LSB-M in the sense that it embeds at most one bit of data in each pixel. As a future work we intend to increase the embedding rate by embedding more than one bit in each candidate pixel by using the improved versions of LSB-M algorithm.

References

1. Cancelli G, Cox IJ, Doerr G (2008) Improved LSB matching steganalysis based on the amplitude of local extrema, in IEEE International Conference on Image Processing, October
2. Cancelli G, Doerr G, Cox IJ, Barni M (2008) A comparative study of +1 steganalyzers, IEEE Int. Workshop. on Multimedia Signal Processing, IEEE Workshop on Multimedia Signal Processing, (MMSP)
3. Chang CC, Tseng HW (2004) A Steganographic method for digital images using side match. Pattern Recogn Lett pp. 1431–1437
4. Chen P, Wu W (2009) A Modified Side Match Scheme for Image Steganograph. International Journal of Applied Science and Engineering, pp. 53–60
5. Dumitrescu S, Wu X, Wang Z (2003) Detection of LSB Steganography via sample pair analysis. IEEE Trans Signal Process 51(7):1995–2007
6. Goljan M, Fridrich J, Holotyak T (2006) New blind steganalysis and its implications. In Security, Steganography, and Watermarking of Multimedia Contents VIII, ser. Proceedings of SPIE, Vvol. 6072, pp. 607 201–1
7. Harmsen JJ, Pearlman WA (2003) Steganalysis of Additive Noise Modelable Information Hiding, Proc. SPIE Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents, Vol. 5020, pp. 131–142
8. Huang F, Li B, Huang J (2007) Attack LSB matching steganography by counting alteration rate of the number of neighbourhood gray levels. Proc IEEE ICIP 1:401–404
9. Ker AD (2005) Steganalysis of LSB matching in grayscale images. IEEE Signal Process Lett 12(6):441–444
10. Li X, Yang B, Cheng D, Zeng T (2009) A generalization of LSB matching. IEEE Signal Process Lett 16 (2):69–72
11. Liu TC, Huang CC (2007) Lossless Information Hiding Scheme Based on Pixels Complexity Analysis, Proceedings of Third International Conference on Signal Image Technology & Internet-based Systems (SITIS 2007), Shanghai, China, pp. 934–941
12. Liu C, Li X, Lu X, Yang B (2009) A content-adaptive approach for reducing embedding impact in steganography. In Proc IEEE ICIP
13. Luo W, Huang F, Huang J (2010) Edge adaptive image steganography based on LSB matching revisited. IEEE Trans Inform Forens Sec 5(2):201–214

14. Maniccam SS, Bourbakis N (2004) Lossless compression and information hiding in images. Pattern Recogn 37:475–486
15. Mielikainen J (2006) LSB matching revisited. IEEE Signal Process Lett 13(5):285–287
16. Omoomi M, Samavi S, Dumitrescu S (2010) An efficient high payload ±1 data embedding scheme. J Multimed Tool Appl 54(2):201–218
17. Sabeti V, Samavi S, Mahdavi M, Shirani S (2007) Steganalysis of pixel-value differencing steganographic method. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp. 292–295
18. Sabeti V, Samavi S, Mahdavi M, Shirani S (2010) Steganalysis and payload estimation of embedding in pixel differences using neural networks. Pattern Recogn 43:405–415
19. United States Department of Agriculture (2002) Natural resources conservation service photo gallery. [Online]. Available: http://photogallery.nrcs.usda.gov
20. Wu DC, Tsai WH (2003) A steganographic method for images by pixel-value differencing. Pattern Recognit Lett 24:1613–1626
21. Yang CH, Weng CY, Wang SJ, Sun HM (2008) Adaptive data hiding in edge areas of images with spatial LSB domain systems. IEEE Trans Inform Forens Sec 3(3):488–497
22. Zhang J, Cox IJ, Doerr G (2007) Steganalysis for LSB matching in images with high-frequency noise. In Proceedings of the IEEE Workshop on Multimedia Signal Processing, pp. 385–388, October

Vajiheh Sabeti received her B.S. degree (summa cum laude) in Software Engineering in 2004 and her M.Sc. degree (honor) in Computer Architecture in 2007, from the Electrical and Computer Engineering department of Isfahan University of Technology (IUT), Isfahan, Iran. Miss Sabeti has been a Ph.D. candidate since autumn of 2007 at the ECE department of IUT where she is working in the field of image processing. Her research interests are softcomputing, image processing, and watermarking.

Shadrokh Samavi is a Professor of Computer Engineering at Isfahan University of Technology, Iran and an Adjunct Professor at the ECE department of McMaster University, Canada. He completed a B.S. degree in Industrial Technology and received a B.S. degree in Electrical Engineering at California State University, a M. S. degree in Computer Engineering at the University of Memphis and a Ph.D. degree in Electrical Engineering at Mississippi State University, U.S.A. Professor Samavi is a Registered Professional Engineer (PE), USA. He is also a member of IEEE and a member of Eta Kappa Nu and Tau Beta Pi honor societies. Shadrokh Samavi's research interests are in the areas of image processing and hardware implementation and optimization of image processing algorithms. He is also interested in watermarking of images and its related subjects.

Shahram Shirani received his Bachelor of Engineering degree in electrical engineering from Isfahan University of Technology, Iran, and his Master of Science (with honor) degree in biomedical engineering from Amirkabir University of Technology, Iran, and his Ph.D. degree in electrical engineering from University of British Columbia, Canada, in 1989, 1994, and 2000, respectively. Since July 2000, he has been with the Department of Electrical and Computer Engineering, Mc-Master University, where he is now an Associate Professor. His research interests are mainly focused on image and video processing, multimedia compression and communications, medical image processing and hardware architectures for image and video processing. He has published more than 160 journal and conference papers. He is a Senior Member of IEEE, a Member of Technical Program Committee of ICIP, a Member of Technical Program Committee of ICASSP, and a Professional Engineer.