International Journal of Electrical and Computer Engineering (IJECE)
Vol. 4, No. 4, August 2014, pp. 573~584
ISSN: 2088-8708


# An Adaptive Steganography Scheme Based on Visual Quality and Embedding Capacity Improvement

**Mojtaba Bahmanzadegan Jahromi*, Karim Faez****
* Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Qazvin, Iran
** Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran

**Article Info**

***Article history:***

Received Apr 21, 2014
Revised May 30, 2014
Accepted Jun 16, 2014

***Keyword:***

Adaptive steganography
Least-significant-bit substitution
Pixel-value differencing

**ABSTRACT**

In this paper, a steganography technique using LSB substitution and PVD method is presented as an adaptive scheme in the spatial domain. Our method partitions the grayscale image into several non-overlapping blocks with three consecutive pixels. The embedding algorithm can both replace the secret data with the LSBs of the middle pixel and embed it in the difference values between the middle pixel and its two neighboring pixels of the cover-block. The number of secret bits is determined adaptively based on the range divisions for embedding in the difference value. We define a new range division on gray level which takes into account a larger embedding capacity for bits. After the embedding, the proposed method detects the pixels which are sensitive to hyper distortion. Then, the embedding process will be repeated to produce insignificant visual distortion in those pixels. Our experimental results demonstrate that this iterative steganography scheme prevents significant visual distortion into stego-image. The generated PSNR values are higher than the corresponding values of the most commonly used methods, discussed in this study. Furthermore, the experimental results show that the hiding capacity increased enormously when the proposed range division is used. Finally, we illustrate that the method can pass RS and steganalysis detector attacks.



***Corresponding Author:***

Mojtaba Bahmanzadegan Jahromi,
Faculty of Computer and Information Technology Engineering,
Qazvin Branch Islamic Azad University,
Barajin University Road, Qazvin, Iran.
Email: bahmanzadegan@qiau.ac.ir

## 1. INTRODUCTION

In recent years, data security has become one of the most important issues of human societies due to increased data transmission over computer networks [1]. In this context, secret communication science has been presented to increase information security. Steganography is one of the most important techniques used to provide safe communication and hide secret messages [2]. It has been used since ancient times and then turned into an integral part of the digital era after the development of computers [3]. In fact, steganography is carried out by hiding secret messages into a cover media such as text, image, video, etc. In this paper, an image has been used as the cover-image and the result image of the embedding process is named stego-image. In contrast, steganalysis is the science of finding such hidden messages [4].

Visual quality, embedding capacity, and information security are three features investigated by researchers in steganographic evaluation. Moreover, the purpose of steganography is to render secret messages imperceptible. Although it is impossible to achieve excellent results for all these features in a steganography scheme, acceptable levels can be realized. However, none of them should be prioritized over the other two.

In a steganography method, the secret message hiding process involves two basic steps. The first step is embedding and the second is extracting. In the embedding phase, a cover image is chosen based on the steganography algorithm performance, suitably for secret data and the cover characteristics. The steganography scheme specifies appropriate regions of the cover-image and then embeds the secret message into them. Then, the resulting stego-image is forwarded to the receiver. In the extracting phase, the receiver gives the stego to the extraction function and the secret message will be extracted.

In view of putting secret data into the cover-image, existing steganography methods can be divided into two general groups: (i) Transform domain methods, (ii) Spatial domain methods. The transform domain approaches [5]-[9] convert the cover-image into another domain (like the frequency) first, and then embed the secret message into the transformed coefficients. Although these methods adequately resist against steganalysis attacks, their time complexity is high. In proposed spatial domain methods [10]-[21], the secret data is embedded directly into the cover-pixel value. Due to their low time complexity, these methods are quite common. Although these techniques provide high embedding capacity, the stego-image quality is consumedly reduced at higher hiding capacities. Increase in their embedding capacity while maintaining acceptable quality appears to have become a challenge. Here, we proposed a method to provide good stego quality at higher capacities.

Spatial domain approaches can be divided into three main categories: (i) LSB replacement methods, (ii) Edge adaptive methods, (iii) Hybrid methods. In LSB replacement methods, LSBs of pixels are used to hide secret messages [10]-[13]. The LSB substitution is a well-known technique in this group. Although these methods provide a large embedding capacity, they are very susceptible to steganalytic attacks. In general, eyes detect changes in smooth areas with additional capabilities than edge areas based on human visual system (HVS) characteristics. The second type of spatial domain methods [14]-[17] is the edge adaptive methods using this feature. The PVD method [14] is an example of this group. This method computes the secret data embedding capacity by the difference values between each pixel and its neighboring pixels. Although the edge adaptive methods produce high visual quality, their hiding capacity is low compared with other techniques like LSB and the like. Recently in [18]-[21], the LSB substitution and the edge adaptive methods have been combined. Thus, the third type of spatial domain methods is the hybrid methods which hide in both LSBs and difference values between neighboring pixels. These are the approaches with which researchers have tried to obtain acceptable values for each of the three features important in the steganographic evaluation. The scheme proposed by Khodaei *et al.* [19] offers acceptable visual quality and large embedding capacity of stego-image among other methods in this category.

In this paper, a hybrid technique is proposed using LSB replacement and PVD method. The scheme both enhances the hiding capacity and improves the visual quality at higher capacities. Three different advantages of our scheme in comparison with other methods are as follows:

1. By defining new range division on gray level R=[0,255], we are able to embed large secret bits, higher than the Yang *et al.'s* [15], the Wu *et al.'s* [18] and the Khodaei *et al.'s* [19] techniques.
2. Our scheme prevents large difference values between the cover image and its stego-image's pixels, unlike the Khodaei *et al.'s* [19] technique. Thus, it produces insignificant visual distortion in hidden messages.
3. Finally, our algorithm resists against RS steganalysis attack, unlike the Wu *et al.'s* [18] and the Yang *et al.'s* [15] methods.

The remainder of this paper is organized as follows. In Section 2, we review two well-accepted data embedding schemes using the LSB replacement and the PVD method. Then, in Section 3, we present our proposed method. Experimental results will be described and compared with the Yang *et al.'s* [15], the Wu *et al.'s* [18] and the Khodaei *et al.'s* [19] methods in Section 4. Finally, the conclusion follows in Section 5.

## 2. ANALYSIS OF RELEVENT APPROACHES

Here, we will describe two most commonly used methods in the next subsections. These techniques use the combination of LSB replacement and the PVD method in spatial domain.

### 2.1. Steganographic Method using LSB Substitution and PVD

Wu *et al.* [18] proposed a steganographic method for grayscale images in 2005. Suppose that $R_j = [l_j, u_j]$ and (j = 1,2, …,5) where $l_j$ and $u_j$ are the lower and upper bound values and $|R_j|$ is the length of the $R_j$ range. This approach divides the R=[0,255] range to the sub-ranges $R_1 = [0, 15]$, $R_2 = [16, 31]$, $R_3 = [32, 63]$, $R_4 = [64, 127]$ and $R_5 = [128, 255]$. The *div* defines the location of range divisions where $div \in \{15,31,63,127\}$. Then, the sub-ranges which are lower than *div* fall into the 'lower level' and the other

sub-ranges are located in the 'higher level'. For instance, Figure 1 shows a division on the gray level when $div = 31$.

At first, this method partitions the cover-image into several non-overlapping blocks having two consecutive pixels, denoted by ( $p_i, p_{i+1}$). For each block, the difference value $d_i$ is calculated by $d_i = |p_i - p_{i+1}|$ where $i$ is the block number. Afterwards, two scenarios may occur in the secret data embedding process:

*Case 1* (If the difference value $d_i$ falls into the lower level): in this case, the LSB replacement method is used to embed 6bits of secret data.

*Case 2* (If the difference value $d_i$ is placed in the higher level): the PVD method is used to embed the secret bits. In this case, the number of embedded bits $t_i$ is calculated by $t_i = \lfloor log_2(|R_j|) \rfloor$.

### 2.2. Adaptive Steganographic Method using LSB Substitution and PVD

Khodaei *et al.* [19] introduced an adaptive steganographic method for grayscale images in 2011. Suppose that $R_j = [l_j, u_j]$ and $(j = 1,2,\ldots,5)$ where $l_j$ and $u_j$ are the lower and upper bound values. The length of $R_j$ is denoted by $|R_j|$. As demonstrated in Figure 2, this method locates the sub-ranges $R_1 = [0,7]$, $R_2 = [7,15]$, $R_3 = [16,31]$, $R_4 = [32,63]$ and $R_5 = [64,255]$ in two levels, denoted by 'lower level' and 'higher level'. Then, it defines *Type1* and *Type2* divisions on the gray level. In the *Type1* division shown in Figure 2a, the sub-ranges $R_1 = [0,7]$ ، $R_2 = [7,15]$ and $R_3 = [16,31]$ are located in the 'lower level' and the sub-ranges $R_4 = [32,63]$ and $R_5 = [64,255]$ fall into the 'higher level'. It hides $t_j = 3$ bits of secret data in the sub-ranges of the lower level, and conceals $t_j = 4$ bits of secret data in the sub-ranges of the higher level. Also, in *Type2* division shown in Figure 2b, the sub-ranges $R_1 = [0,7]$, $R_2 = [7,15]$, $R_3 = [16,31]$ and $R_4 = [32,63]$ belong to the 'lower level' and $R_5 = [64,255]$ is assigned to the 'higher level'. In this division type, the number of secret bits $t_i$ is calculated in the lower level, by $t_j = \lfloor log_2(|R_j|) \rfloor$ and in the higher level, by $t_j = \lfloor log_2(|l_j|) \rfloor$.

At first, this method partitions the cover-images into several non-overlapping blocks with three consecutive pixels, denoted by $B_i$ where $i$ is the block number. The $k$ and $k \epsilon \{3,4,5,6\}$ define the number of secret bits that can be embedded in the LSBs. Therefore, the embedding capacity will grow as $k$ is increased. The type of range division and the $k$ value should be selected in the data embedding process first. For each cover block, the $k$-bits of secret data are substituted for the $k$-LSB of middle pixel $p_{ic}$ and $p'_{ic}$ is obtained. Then, the values $d_{i1}$ and $d_{i2}$ are calculated by the difference between the middle pixel $p'_{ic}$ and its two neighboring pixels $p_{i1}$and $p_{i2}$. The $R_{j1}$ and $R_{j2}$ ranges to which $d_{i1}$ and $d_{i2}$ belong, are selected from the considered range division. The new difference values $d'_{i1}$ and $d'_{i2}$ are calculated by the number of secret bits $t_{j1}$ and $t_{j2}$ and the upper and the lower bounds of their ranges. Finally, taking the new difference values $d'_{i1}$ and $d'_{i2}$ into consideration, the method produces two values for each pixel. Whereas one of them has much difference compared with the original value of cover-image, the similar value is chosen for its stego-image.

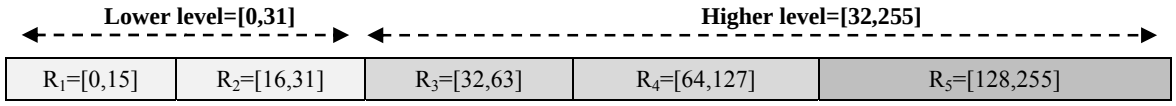



Figure 1. An example of range division on gray level (R=[0,255]) into 'lower level' and 'higher level' in the *Wu et al.* method (*div*=31)

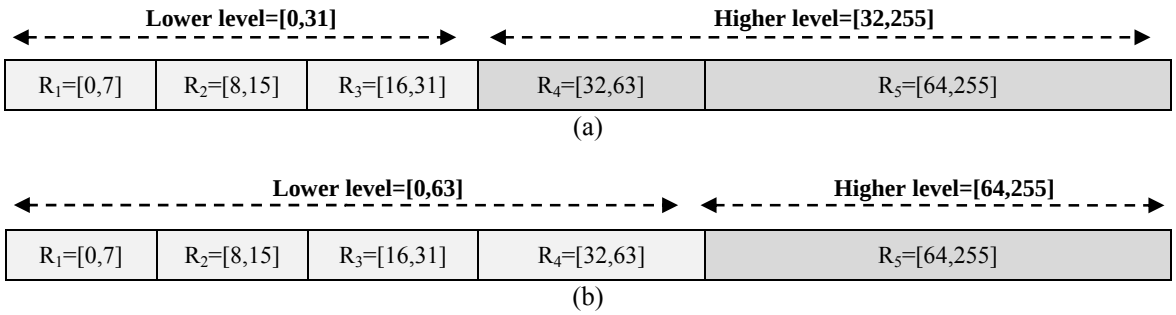



Figure 2. Two range divisions on gray level ($R = [0,255]$) into 'lower level' and 'higher level' in the *Khodaei et al.* method, a) Type1, b) Type2

## 3. THE PROPUSED METHOD

In this section, we will define our proposed method for grayscale images. The aims of proposing this method are finding sensitive pixels to the hyper distortions and extending the data embedding process recursively. Also, the method will be increase the hiding capacity through defining a new range division. Our proposed method is presented in three phases: *(i)* Range division on gray level phase, *(ii)* Data embedding process, (iii) Data extracting process, in the following subsections.

### 3.1. Range Divisions on Gray Level Phase

Here, we consider two range divisions on gray level, *Type1* and *Type2*. The *Type1* division is used to compare the proposed method with the other related methods based on the visual quality, specified by the Khodaei *et al.* [19] method. Additionally, the *Type2* division is defined as our proposed range division in order to achieve a larger embedding capacity. First, suppose that $R_j = [l_j, u_j]$ and $(j = 1,2,\dots,5)$ where $l_j$ and $u_j$ are the lower and upper bound values. The width of $R_j$ is denoted by $|R_j|$.

#### 3.1.1. Type 1 Division

Here, the sub-ranges $R_1 = [0,7]$, $R_2 = [7,15]$, $R_3 = [16,31]$ and $R_4 = [32,63]$ are put in the 'lower level' and the sub-range $R_5 = [64,255]$ in the 'higher level', according to the Khodaei *et al.*'s division shown in Figure 2b. In this type of range division, the number of secret bits $t_i$ is calculated by $t_j = |log_2(|R_j|)|$ in the sub-ranges of the lower level and by $t_j = |log_2(|l_j|)|$ in the higher level. So, the number of the embedded bits for $R_j$ where $(j = 1,2,\dots,5)$ will be $t_1 = 3$, $t_2 = 3$, $t_3 = 4$, $t_4 = 5$ and $t_5 = 6$.

#### 3.1.2. Type 2 Division

We assign the sub-range $R_1 = [0,7]$ to the 'lower level', the sub-ranges $R_2 = [7,15]$, $R_3 = [16,31]$ and $R_4 = [32,63]$ to the 'middle level', and the sub-range $R_5 = [64,255]$ to the 'higher level', in the proposed *Type2* division shown in Figure 3. Here, the number of secret bits $t_i$ is calculated by $t_j = |log_2(|R_j|)|$ in the sub-range of the lower level, by $t_j = |log_2(|l_j|)|+1$ in the middle level, and by $t_j = |log_2(|l_j|)|$ in the higher level. Thus, it will be $t_1 = 3$, $t_2 = 4$, $t_3 = 5$, $t_4 = 6$ and $t_5 = 6$ for $R_j$ and $(j = 1,2,\dots,5)$.

### 3.2. Data Embedding Process

First, one of the *Type1* or the *Type2* divisions should be selected. The flow diagram of the proposed steganography scheme is illustrated in Figure 4. We suggest 12 steps for the proposed data embedding process in the following procedure:

*Step 1:* A grayscale image is partitioned into non-overlapping bocks having three consecutive pixels. The first pixel, middle pixel and second pixel are denoted by $(p_{i1}, p_{ic}, p_{i2})$, where $i$ is the number of the block. $S$ also denotes the secret data.

*Step 2:* Consider the $k$ value where $k\epsilon\{3,4,5,6\}$ is the number of the secret bits that can be embedded in LSBs. Then, the embedding capacity is increased by the higher value of $k$. Thus, $p'_{ic}$ is obtained by putting *k-leftmost* bits of the binary secret data ($S$) into *k-rightmost* bits of LSBs of $p_{ic}$.

*Step 3:* Compute the difference value $d$ between $LSB_i$ and $s_{ic}$ using Eq. (1).

$$d = LSB_i - s_{ic} \quad (1)$$

where $LSB_i$ is the decimal value of *k-rightmost* LSBs of $p_{ic}$ and $s_{ic}$ is the decimal value of *k-leftmost* bits of $S$.

*Step 4:* Use optimal pixel adjustment process (OPAP) [15] and alter the value of $p'_{ic}$, as shown in Eq. (2).

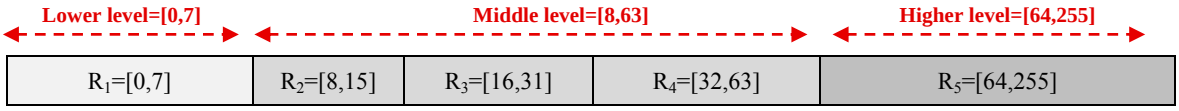



Figure 3. The new proposed *Type2* division on gray level ($R = [0,255]$) containing the 'lower level', the 'middle level' and the 'higher level

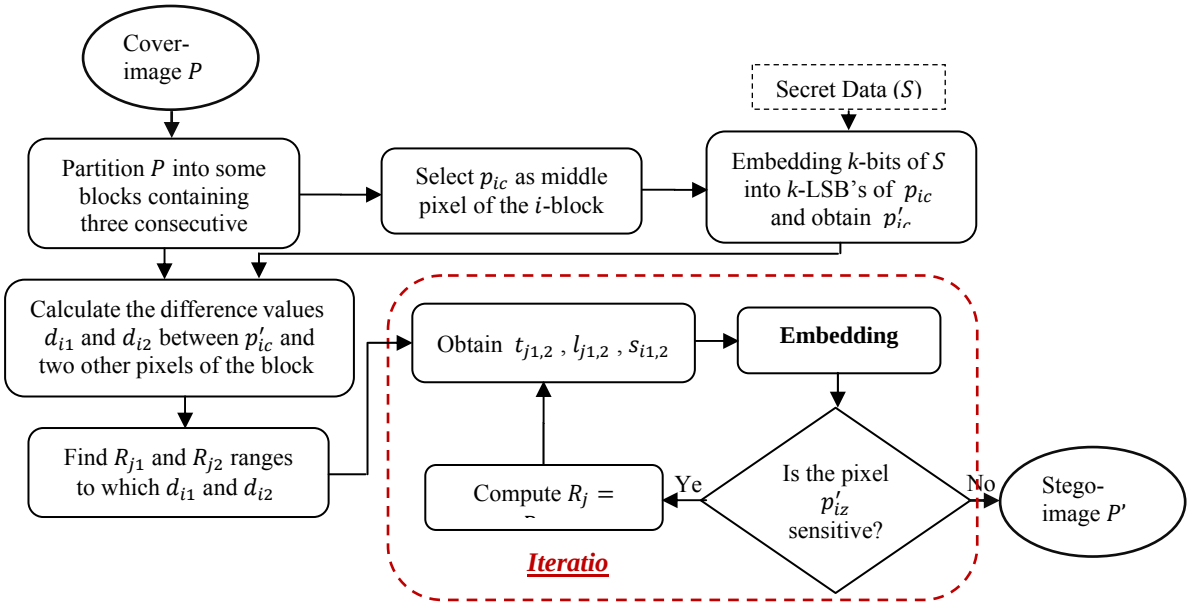



Figure 4. Block diagram of the iterative embedding process

$$p'_{ic} = \begin{cases} p'_{ic} + 2^k & if\ d > 2^{k-1}\ \ and\ \ 0 \le p'_{ic} + 2^k \le 255 \\ p'_{ic} - 2^k & if\ d < -2^{k-1}\ \ and\ \ 0 \le p'_{ic} - 2^k \le 255 \\ p'_{ic} & otherwise \end{cases} \quad (2)$$

*Step 5:* Calculate the difference values, $d_{i1}$ and $d_{i2}$, between the middle pixel $p'_{ic}$ and its two neighboring pixels $p_{i1}$ and $p_{i2}$ of the cover-block by Eq. (3).

$$d_{i1} = |p_{i1} - p'_{ic}| \quad , \quad d_{i2} = |p_{i2} - p'_{ic} \quad (3)$$

*Step 6:* Find the $R_{j1}$ and $R_{j2}$ ranges of the range division in question if $d_{i1}$ and $d_{i2}$ belong to the ranges.
*Step 7:* Obtain the numbers of binary secret bits $t_{j1}$ and $t_{j2}$ and find the lower bounds $l_{j1}$ and $l_{j1}$ of the corresponding $R_{j1}$ and $R_{j2}$ ranges.
*Step 8:* Select $t_{j1}$ and $t_{j2}$ bits of $S$, and transform the two bit sequences to decimal values $s_{i1}$ and $s_{i2}$.
*Step 9:* Calculate the new difference values $d'_{i1}$ and $d'_{i2}$ using Eq. (4).

$$d'_{i1} = l_{j1} + s_{i1} \quad , \quad d'_{i2} = l_{j2} + s_{i2} \quad (4)$$

*Step 10:* Calculate $p'_{i1}$ and $p'_{i2}$ values for the first and the third pixels in the block, by obtaining the difference between the original value and the new value of them using Eq. (5).

$$p'_{i1} = \begin{cases} 1.\ p'_{ic} - d'_{i1} & if\ p'_{ic} \ge d'_{i1} \\ 2.\ p'_{ic} + d'_{i1} & if\ p'_{ic} < d'_{i1} \end{cases},\ p'_{i2} = \begin{cases} 1.\ p'_{ic} - d'_{i2} & if\ p'_{ic} \ge d'_{i2} \\ 2.\ p'_{ic} + d'_{i2} & if\ p'_{ic} < d'_{i2} \end{cases} \quad (5)$$

*Step 11:* Detect the sensitive pixels: four conditions will be checked for $p'_{iz}$in this step where $z\epsilon\{1,2\}$ and $(p'_{i1}, p'_{i2})$.

I. If $(p'_{iz} < 0)$: the under-flow problem occurred.
II. If $(p'_{iz} > 255)$: the over-flow problem happened.
III. If $(d'_{iz} \notin R_j)$: the difference value between stego and its cover is wrong.
IV. If $(l_{jz} > 0)$: the embedding process can be repeated.

Thus, if at least one of the first three conditions as well as the last condition are satisfied, or $((p'_{iz} < 0) \vee (p'_{iz} > 255) \vee (d'_{iz} \notin R_j)) \wedge (l_{jz} > 0)$ is true, the $p'_{iz}$ is sensitive on the embedded secret bits and the embedding procedure can also be repeated.
*Step 12:* Define two cases for $p'_{iz}$ where $z\epsilon\{1,2\}$ and $(p'_{i1}, p'_{i2})$ as follows

I. Case 1 ( $p'_{iz}$ is sensitive )

- Find $R_{jz-1}$ range as the previous range through $R_{jz}$ of the selected division.
- Redo embedding process from step 7 to 11 for $p'_{iz}$.

II. Case 2 ( $p'_{iz}$ is insensitive )

- Consider the value of $p'_{iz}$ calculated in Step10. When $p'_{iz}$ falls into the under-flow or in the over-flow problem, the second value should be calculated in Step10.

Finally, the stego-block will be produced. Please note that the above procedure should be repeated for each cover-block. Thus, the secret message will be embedded completely in the cover image and its stego-image will be produced.

### 3.3. Data Extracting Process

For secret message extraction, suppose that the type of range division and the *k* value used in the embedding phase are available. Meanwhile please note that their values have been considered based on the purpose of hiding. This procedure is started with dividing the stego-image into non-overlapping blocks each of which 3×3 pixels. For each block as *i*-block, *k*-bits of the middle pixel $p'_{ic}$ should be selected. This sequence of bits is placed in the rightmost of secret bits ($S$). Then, we calculate the difference values $d'_{i1}$ and $d'_{i2}$ between the middle pixel $p'_{ic}$ and its two neighboring pixels $p'_{i1}$ and $p'_{i2}$ of the stego-block, using Eq. (6).

$$d'_{i1} = |p'_{i1} - p'_{ic}| \quad , \quad d'_{i2} = |p'_{i2} - p'_{ic}| \tag{6}$$

The $R_{j1}$ and $R_{j2}$ ranges to which $d'_{i1}$ and $d'_{i2}$ belong are. The lower bound values $l_{j1}$ and $l_{j2}$ and the number of embedded bits $t_{j1}$ and $t_{j2}$ are chosen, considering these ranges. Therefore, the secret values are calculated by Eq. (7).

$$s_{i1} = d'_{i1} - l_{j1} \quad , \quad s_{i2} = d'_{i2} - l_{j2} \tag{7}$$

In the final step, $s_{i1}$ and $s_{i2}$ values are converted into their binary sequences based on the number of the bits $t_{j1}$ and $t_{j2}$. Then, these binary sequences should be added to the rightmost of the secret bits ($S$). Finally, the secret message will be extracted properly without any distortion.

### 3.4. Simple Example of the Proposed Method

In this section, we will implement a simple example of the proposed method. The next subsections present the data embedding and extracting processes. Thus, we will demonstrate that secret bits are properly embedded and extracted using our proposed method.

***I. Data embedding process*: *Step1***, we will use *Type1* division and $k = 3$ for this example. The secret message is $S = (1101000011010)$ and the selected block contains the three pixels $p_{i1} = 109$ and $p_{ic} = 44$ and $p_{i2} = 3$. ***Step2***, the *k-leftmost* of secret bits should be replaced with *k-LSBs* of $p_{ic}$. The binary value of the middle pixel is $p_{ic} = 44 = (101100)_2$, so the decimal value will be $p'_{ic} = (101110)_2 = 46$. ***Step3***, the difference value $d$ is calculated between $LSB_i = 4$ that is the decimal value of *k-LSBs* of $p_{ic}$ and $s_{ic} = 6$ that is the decimal value of *k-leftmost* of the secret bits by $d = 4 - 6 = -2$. ***Step4***, we obtain $p'_{ic} = 46$ using OPAP [21]. ***Step5***, the difference value is calculated between $p'_{ic}$ and the other two pixels $p_{i1}$ and $p_{i2}$ by $d_{i1} = |109 - 46| = 63$ and $d_{i2} = |3 - 46| = 43$. ***Step6***, the value $d_{i1} = 63$ is in $R_4$ and $d_{i2} = 43$ is in $R_4$ considering the *Type1* division. ***Step7***, the values $t_{j1} = 5$ and $t_{j2} = 5$ that are the number of secret bits and $l_{j1} = 32$ and $l_{j2} = 32$ that are the lower bound values of the corresponding ranges are obtained. ***Step8***, the five bits of the secret message is selected and converted into the decimal value as $s_{i1} = (10000)_2 = 16$. In addition, five of secret bits are transformed to the decimal value as $s_{i2} = (11010)_2 = 26$. ***Step9***, the new difference values is calculated using (4) as $d'_{i1} = 32 + 16 = 48$ and $d'_{i2} = 32 + 26 = 48$. ***Step10***, the new values $p'_{i1} = 94$ and $p'_{i2} = -12$ are calculated. ***Step11***$(z = 1)$, the $p'_{i1}$ is an insensitive pixel because of the $(p'_{i1} > 0)$ and $(p'_{i1} < 255)$ and $(d'_{i1} \in R_4)$ conditions. ***Step12***$(z = 1)$, $p'_{i1} = 94$ should be considered in accordance with the first stage of this step. ***Step11***$(z = 2)$, based on two conditions $(p'_{i2} < 0)$ and $(l_{j2} > 0)$, the $p'_{i2}$ is a sensitive pixel. ***Step12***$(z = 2)$, therefore, we consider $R_{4-1} = R_3$ range of *Type1* division based on the second case of this step. Then, the embedding process repeats step 7 to 11 for $p'_{i2}$. ***Step7***, the number of secret bits and the lower bound value are $t_{j2} = 4$ and $l_{j2} = 16$ based on $R_3$ range of *Type1* division. ***Step8***, the four bits of the secret message is chosen and converted into the decimal value as $s_{i2} = (1101)_2 = 1$. ***Step9***, the difference value is calculated as $d'_{i2} = 16 + 13 = 29$. ***Step10***, the new value $p'_{i2} = 17$ is calculated. ***Step11***, there are three conditions $(p'_{i2} > 0)$ and $(p'_{i2} < 255)$ and $(d'_{i2} \in R_3)$. So, $p'_{i2}$ is an insensitive pixel. ***Step12***, based on the first case of this step, the final

value is $p'_{i2} = 17$. Finally, the secret message $S = (110100001101)$ is embedded properly in the three pixels $p'_{i1} = 94$ and $p'_{ic} = 46$ and $p'_{i2} = 17$ of the stego-block.

***II. Data extracting process*:** We used *Type1* division and $k = 3$ and divided the cover image into some blocks having three pixels in the embedding phase. At first, the value of the middle pixel $p'_{ic}$is converted into binary value as $p'_{ic} = (101110)_2$. The *three LSBs* of $p'_{ic}$ are selected as the *three rightmost* bits of secret data. Then, we calculate the difference values $d'_{i1} = |94 - 46| = 48$ and $d'_{i2} = |17 - 46| = 29$ by Eq. (6). Based on the *Type1* division and the $R_4$ range to which $d'_{i1}$ belongs, the values $l_{j1} = 32$ and $t_{j1} = 5$ are obtained. So, $s_{i1}$ is $s_{i1} = 48 - 32 = 16$ by Eq. (7) and its binary value is $s_{i1} = (10000)_2$. Moreover, $t_{j2} = 4$ and $l_{j2} = 16$ are considered using the *Type1* division and the $R_3$ to which $d'_{i2}$ belongs. Based on Eq. (7), the value of $s_{i2} = 29 - 16 = 13$ is $s_{i2} = (1101)_2$, when $t_{j2} = 4$. The $s_{i1}$ and $s_{i2}$ bit sequences are added to the secret bits (S). Finally, we could extract the secret message $S = (110100001101)$, correctly.

## 4. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we demonstrate the effectiveness of our proposed scheme compared with the Wu *et al.'s* [18], the Yang *et al.'s* [15] and also the Khodaei *et al.'s* [19] methods. We present some experimental results obtained using 10 cover images with 512×512 image resolutions. All the cover images have been transformed to grayscale images. Cover images include *Baboon*, *Barbara*, *Boat*, *Cameraman*, *Lena*, *Livingroom*, *Peppers*, *Pirate*, *Tiffany* and *Zelda*. Furthermore, the secret bits are produced by a Random Number Generator (RNG). Generally, visual quality, hiding capacity and information security are used in evaluations of a given steganography algorithm. The proposed method provides high data security with higher visual quality while its embedding capacity will be larger than the well-known methods.

The Peak Signal to Noise Ratio (PSNR) value is used to evaluate the distortions of stego-image. We compute the PSNR value in dB by Eq. (8), where the Mean Square Error (MSE) is calculated, as shown in Eq. (9). In Eq. (9), *m* is the shared size of cover *P* and stego *P'* images.

$$PSNR = 10 \times Log\left(\frac{(255)^2}{MSE}\right) \tag{8}$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(p_i - p'_i)^2 \tag{9}$$

Furthermore, let $E$ be the total bits of an embedded message into a stego-image. As shown in Eq. (10), we calculate $E_{bpp}$as the average capacity in bit per pixel($bpp$), where $S$ is the number of secret bits and $m$ is the size of the cover-image.

$$E_{bpp} = S/m \tag{10}$$

### 4.1. Visual Quality

Here, we will measure the visual quality of stego-images produced by our proposed method using subjective and objective means of measurement. In addition, the number of sensitive pixels will be calculated to justify the effectiveness of our proposed method to provide better visual quality.

#### 4.1.1. Subjective Measurement

The secret message imperceptibility to the human eye is the main goal of all steganography techniques. In other words, the human eye, as a means of as a subjective measurement, should be unable to notice the secret message in cover. Our first test case measured the stego-image by the vision system. We embedded the maximum secret data using the *Type1* division and various *k* values on *Peppers* cover-image. Figure 5a and Figure 5b show the cover and its stego. Moreover, the difference between the selected region of the stego and the corresponding region of its cover is presented in Figure 5c.

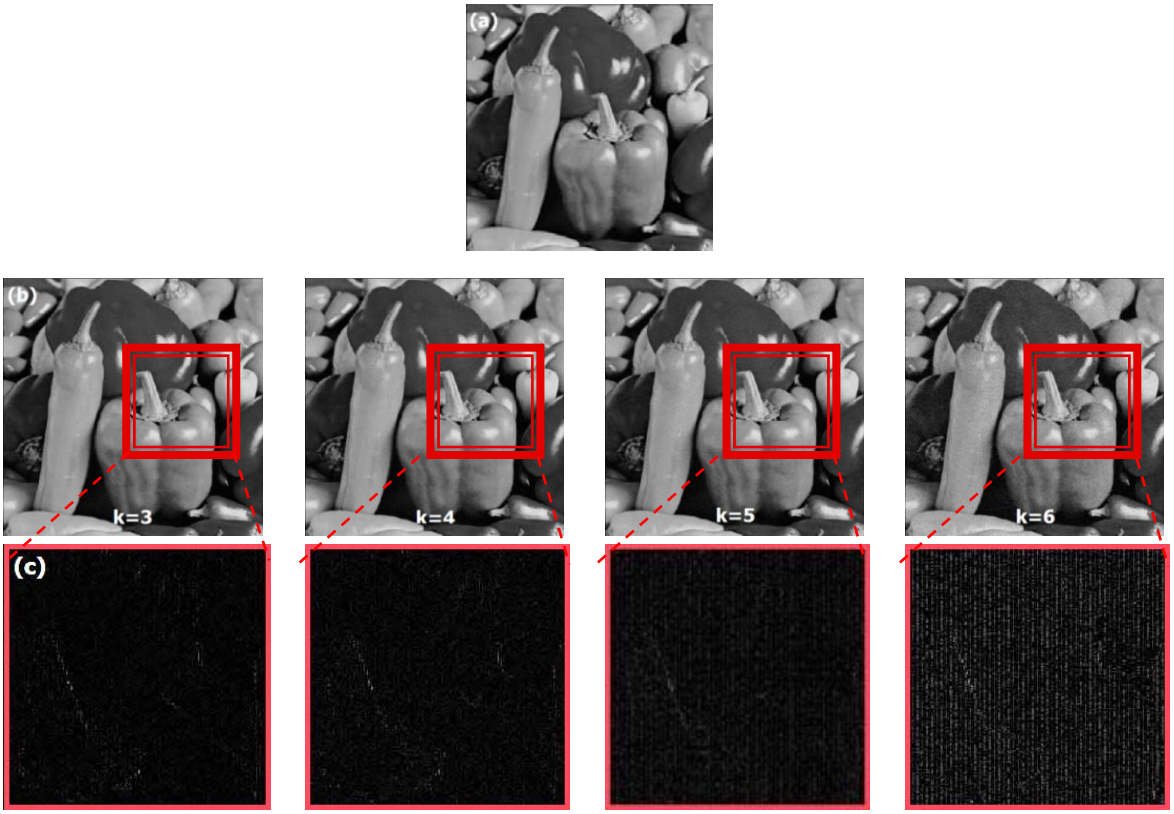



Figure 5. *Stego-image quality analysis*: Embedding the secret bits by using the proposed method with *type1* division and $k = 3$, a) cover-image, b) stego-images, c) difference between the selected region of stego-image and its corresponding cover-image region (***k=3*: $PSNR = 38.95$ dB, $Capacity = 806058$ bits. ***k=4*: $PSNR = 36.64$ dB, $Capacity = 896273$ bits. ***k=5*: $PSNR = 33.46$ dB, $Capacity = 1002440$ bits. ***k=6*: $PSNR = 32.61$ dB, $Capacity = 1157767$ bits

Unlike the Khodaei *et al.'s* [19] technique, our scheme prevents large difference values between the cover and its stego-image's pixels. Thus, the emergence of significant visual distortion is prevented through the proposed method, and the secret message is invisible as tested by the human visual system (HVS).

### 4.1.2. Objective Measurement

We experiment our proposed method using the *Type1* and *Type2* divisions and different *k* values on the cover images. Table 1 presents the number of the sensitive pixels and the PSNR values of the stego-images. The proposed method has found the sensitive pixels to prevent high difference values. As per an objective measurement, the results show that the PSNR values are higher than 30 (dB). Then, its embedding process was modified on these sensitive pixels. Of course, the increased number of the sensitive pixels using the *Type2* division is because of its higher embedding capacity compared with the *Type1* division.

In Figure 6a, we compared the PSNR values of our proposed method with the Wu *et al.'s* [15] and Yang *et al.'s* [18] and also Khodaei *et al.'s* [19] methods where the x-axis presents the stego-image and the y-axis shows the PSNR value. As demonstrated in this figure, the mean PSNR value (37.65 dB) for the stego-images produced by the Adaptive steganography method using LSB replacement and PVD (with $div = 63$, $k = 3$) [19] is better than the corresponding values (37.35 dB and 37.33 dB) of the steganography methods using LSB replacement and PVD [18] and Edge adaptive LSB method (with *division*3-4) [15]. However, the quality of 'Tiffany' and 'Cameraman' stego-images produced by the Adaptive method is reduced due to the large number of sensitive pixels in these images (calculated in Table 1). Furthermore, as shown in Table 1, the number of sensitive pixels is low in '*Zelda*' and also in '*Boat*' using *Type1*. Thus, the quality of them has increased slightly. Based on Figure 6a, the PSNR values of our method (with *Type1* division, $k = 3$) is also more than the Adaptive steganography method. In addition, the quality of 'Baboon' stego-image produced by the proposed method soared in comparison with the others due to the large number of sensitive pixels in it. Therefore, we provide a better visual quality of stego-image than these other methods.

Table 1. *Visual quality analysis*: The number of sensitive pixels ($Sens_{pixel}$) and the PSNR ($dB$) values of stego-images produced by the proposed method (with *Type1* and *Type2* division and different *k* values)

| Cover-images | PSNR, dB | Sens, Pixel | PSNR, dB | Sens, Pixel | PSNR, dB | Sens, Pixel | PSNR, dB | Sens, Pixel |
|---|---|---|---|---|---|---|---|---|
| | | | | ***Type1* division** | | | | |
| Baboon | 38.76 | 691 | 35.88 | 728 | 32.49 | 960 | 31.21 | 1518 |
| Barbara | 37.42 | 119 | 34.18 | 85 | 31.47 | 99 | 30.72 | 103 |
| Boat | 37.48 | 22 | 35.27 | 38 | 32.21 | 29 | 31.41 | 69 |
| Cameraman | 39.41 | 272 | 37.94 | 263 | 34.71 | 419 | 32.81 | 1140 |
| Lena | 38.10 | 12 | 36.05 | 11 | 32.61 | 20 | 31.53 | 26 |
| Peppers | 38.95 | 251 | 36.64 | 377 | 33.46 | 701 | 32.61 | 1641 |
| Livingroom | 39.44 | 154 | 37.20 | 174 | 33.24 | 237 | 32.21 | 386 |
| Pirate | 39.75 | 12 | 37.44 | 19 | 33.35 | 16 | 32.41 | 27 |
| Tiffany | 39.22 | 591 | 37.04 | 691 | 33.22 | 820 | 32.37 | 868 |
| Zelda | 39.18 | 4 | 37.05 | 17 | 33.14 | 49 | 31.73 | 245 |
| Average | 38.76 | 212 | 36.46 | 240 | 32.99 | 335 | 31.90 | 602 |
| | | | | ***Type2* division** | | | | |
| Baboon | 38.23 | 3 703 | 35.45 | 3 925 | 32.11 | 4 585 | 30.74 | 4 890 |
| Barbara | 37.01 | 2 238 | 33.66 | 2 822 | 31.06 | 4 152 | 30.29 | 4 760 |
| Boat | 37.03 | 1 945 | 34.82 | 2 654 | 31.83 | 4 520 | 30.94 | 5 390 |
| Cameraman | 39.00 | 1 018 | 37.55 | 1 728 | 34.34 | 4 437 | 32.20 | 5 797 |
| Lena | 37.64 | 1 983 | 35.68 | 2 752 | 32.18 | 4 476 | 31.13 | 5 412 |
| Peppers | 38.51 | 2 143 | 36.30 | 2 840 | 33.04 | 4 510 | 32.17 | 5 541 |
| Livingroom | 38.98 | 2 408 | 36.84 | 3 033 | 32.77 | 4 460 | 31.73 | 5 251 |
| Pirate | 39.29 | 2 288 | 37.01 | 2 904 | 32.89 | 4 478 | 31.93 | 5 368 |
| Tiffany | 38.87 | 2 380 | 36.53 | 3 023 | 32.76 | 4 540 | 31.88 | 5 322 |
| Zelda | 38.78 | 1 769 | 36.49 | 2 602 | 32.69 | 4 493 | 31.26 | 5 571 |
| Average | 38.33 | 2 187 | 36.03 | 2 828 | 32.56 | 4 465 | 31.42 | 5 330 |

### 4.2. Embedding Rate

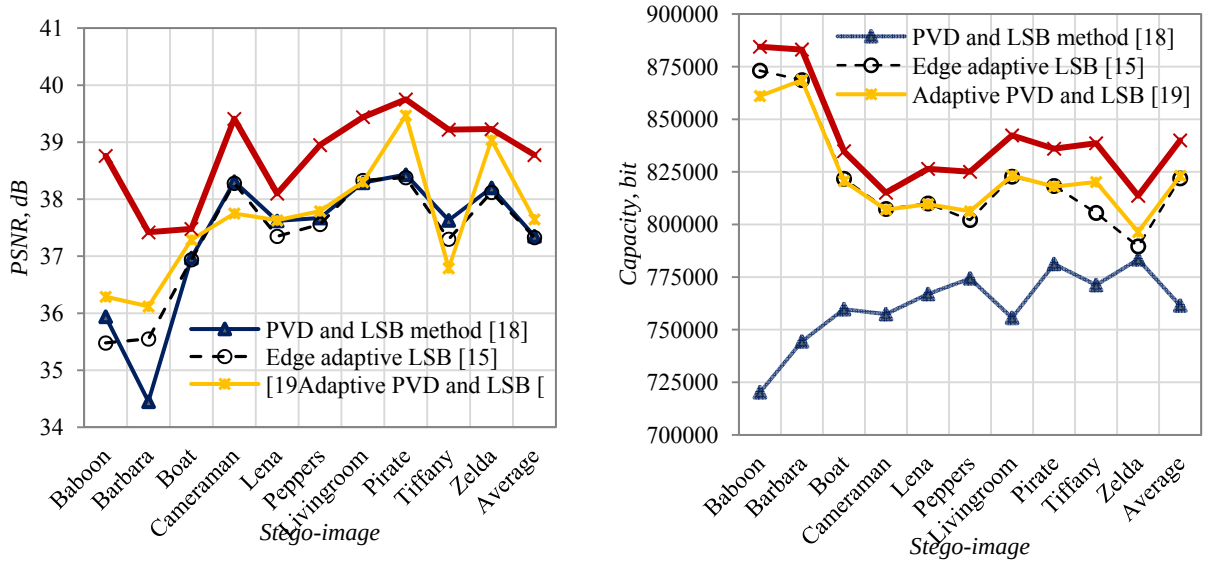
In this test case, we produce stego-images by our proposed method with the *Type1* and *Type2* division and various *k* amounts. Table 2 shows the capacity of embedded secret bits and the average capacity in bit per pixel ($E_{bpp}$) for each range division with different *k* values. As per the experimental results, the capacity and the $E_{bpp}$ values using the *Type2* division are higher than those using the *Type1* division. According to Table 2, the $E_{bpp}$ values are in the [3.037- 4.529] range using *Type1* and in the [3.097- 4.667] range using *Type2*. Also, all $Capacity_{bit}$ average values by *Type2* are higher than their corresponding values using *Type1* (when $k = 4$ and *Type1*: the average Capacity = 912290 bits, when $k = 4$ and *Type2*: the average Capacity = 935101 bits). Thus, the hiding capacity is increased by using our proposed *Type2* division.

In Figure 6b, we compare these results with the Wu *et al.'s* [18], Yang *et al.'s* [15] and Khodaei *et al.'s* [19] methods. This figure shows that the embedding capacity of the proposed method (with *Type2* division, $k = 3$) is larger than the Steganography method using LSB replacement and PVD [18], the Edge adaptive LSB method (with *division* 3-4) [15] and also the Adaptive steganography method using LSB replacement and PVD (with $div = 63$, $k = 3$) [19]. In Figure 6b, the x-axis and the y-axis show the stego-image and the embedding capacity. Thus, we provide a higher embedding capacity than these other methods.

### 4.3. Information Security

Here, the security of the proposed method is tested in terms of RS and steganalysis detector attacks. The RS steganalysis by Fridrich *et al.* [22] in 2001 can show exactly whether a stego-image resists without visual check. This steganalysis method classifies all the stego-image pixels into three groups by using dual statistical methods: the regular group ($R_m$ or $R_{-m}$), the singular group ($S_m$ or $S_{-m}$), and the unusable group. The relation between the percentage of the regular groups and the singular groups is $R_m + R_{-m} \leq 1$ and $S_m + S_{-m} \leq 1$. Here $R_m$ and $S_m$ are percentages with the mask $m$, and $R_{-m}$ and $S_{-m}$ are percentages with the mask $-m$ of the regular and the singular groups, respectively. If $R_m \cong R_{-m}$ and $S_m \cong S_{-m}$, the stego-image will pass the RS attack. Otherwise, the stego-image is detected as a suspicious object.

Figure 7a shows the result of the RS steganalysis (by two masks $m = [0\ 1\ 1\ 0]$ and $-m = [0\ -1\ -1\ 0]$) of the stego-images produced by the proposed method. In this figure, the x-axis presents the embedding capacity percentage and the y-axis shows the percentage of the regular and the singular groups. According to Figure 7a, we were right in expecting that the $R_m$ and $S_m$ relative numbers are respectively equal to those of $R_{-m}$ and $S_{-m}$ (i. e. $R_m \cong R_{-m}$ and $S_m \cong S_{-m}$).

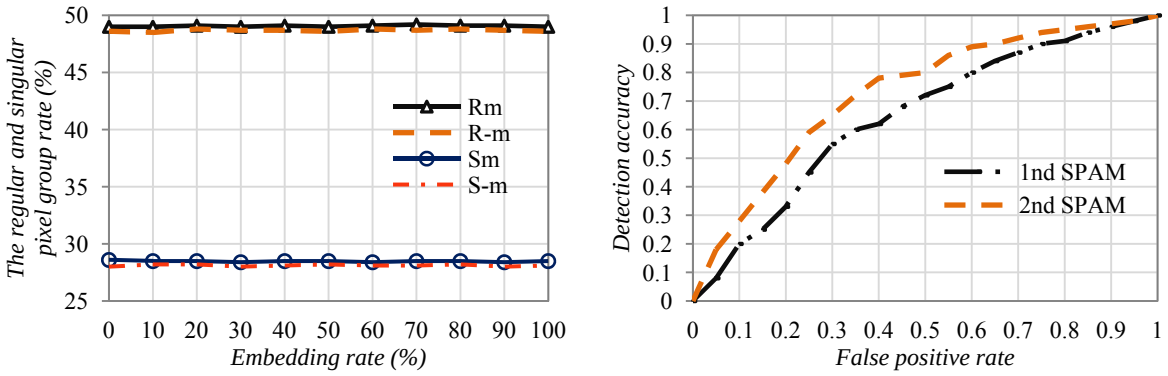



(a) (b)

Figure 6. Comparisons between our proposed method and the steganography method using LSB replacement and PVD, the edge adaptive LSB method (*division*3-4), and the adaptive steganography using LSB replacement and PVD ($div = 63, k = 3$), a) *Visual quality comparison*: (the proposed method with the *Type1* division and $k = 3$), b) *Embedding capacity comparison*: (the proposed method with the *Type2* division and $k = 3$)

Table 2. *Embedding capacity analysis*: The capacity of embedded secret bits ($Capacity_{bit}$) and the average capacity in bit per pixel ($E_{bpp}$) of stego-images produced by the proposed method (with the *Type1* and *Type2* division and different *k* values)

| Cover-images | Capacity, Bit | E, bpp | Capacity, bit | E, bpp | Capacity, bit | E, Bpp | Capacity, bit | E, Bpp |
|---|---|---|---|---|---|---|---|---|
| | | | | ***Type1* division** | | | | |
| Baboon | 860 314 | 3.281 | 950 637 | 3.626 | 1 050 556 | 4.007 | 1 187 380 | 4.529 |
| Barbara | 867 520 | 3.309 | 956 601 | 3.649 | 1 057 542 | 4.034 | 1 198 693 | 4.529 |
| Boat | 819 703 | 3.126 | 908 886 | 3.467 | 1 011 899 | 3.860 | 1 165 134 | 4.444 |
| Cameraman | 806 761 | 3.077 | 895 346 | 3.415 | 994 191 | 3.792 | 1 155 610 | 4.408 |
| Lena | 809 618 | 3.088 | 898 347 | 3.426 | 1 004 140 | 3.830 | 1 160 592 | 4.427 |
| Peppers | 806 058 | 3.074 | 896 273 | 3.419 | 1 002 440 | 3.824 | 1 157 767 | 4.416 |
| Livingroom | 823 061 | 3.139 | 913 089 | 3.483 | 1 017 395 | 3.881 | 1 169 106 | 4.459 |
| Pirate | 818 061 | 3.120 | 907 784 | 3.462 | 1 011 600 | 3.858 | 1 165 940 | 4.447 |
| Tiffany | 819 664 | 3.126 | 909 729 | 3.470 | 1 014 292 | 3.869 | 1 167 518 | 4.453 |
| Zelda | 796 393 | 3.037 | 886 217 | 3.380 | 993 463 | 3.789 | 1 153 358 | 4.399 |
| Average | 822 715 | 3.138 | 912 290 | 3.480 | 1 015 751 | 3.874 | 1 168 109 | 4.455 |
| | | | | ***Type2* division** | | | | |
| Baboon | 884 446 | 3.373 | 974 688 | 3.718 | 1 080 148 | 4.120 | 1 211 679 | 4.622 |
| Barbara | 883 112 | 3.368 | 977 224 | 3.727 | 1 088 385 | 4.151 | 1 223 579 | 4.667 |
| Boat | 834 824 | 3.184 | 930 727 | 3.550 | 1 045 733 | 3.989 | 1 194 801 | 4.557 |
| Cameraman | 815 096 | 3.097 | 910 897 | 3.474 | 1 033 271 | 3.941 | 1 182 964 | 4.512 |
| Lena | 826 429 | 3.152 | 922 421 | 3.518 | 1 041 088 | 3.971 | 1 188 839 | 4.535 |
| Peppers | 825 030 | 3.147 | 921 125 | 3.513 | 1 038 398 | 3.961 | 1 186 898 | 4.539 |
| Livingroom | 842 295 | 3.213 | 937 972 | 3.587 | 1 052 439 | 4.014 | 1 195 459 | 4.560 |
| Pirate | 836 018 | 3.189 | 931 338 | 3.552 | 1 046 856 | 3.993 | 1 192 161 | 4.547 |
| Tiffany | 838 590 | 3.198 | 933 930 | 3.562 | 1 049 521 | 4.003 | 1 194 319 | 4.555 |
| Zelda | 813 666 | 3.103 | 910 696 | 3.474 | 1 031 402 | 3.934 | 1 182 297 | 4.510 |
| Average | 839 950 | 3.204 | 935 101 | 3.567 | 1 050 724 | 4.008 | 1 195 299 | 4.559 |




(a) (b)

Figure 7. *Information security analysis*: a) RS-diagram of stego-images produced by the proposed method, b) receiver operating characteristic curve of steganalyser using the SPAM features with threshold $T = 3$

Figure 7b illustrates the receiver operating characteristic curve, plotted by steganalysis detector using the SPAM features and the support vector machines classifier with a Gaussian kernel [23]. According to Figure 7b, we can see that the security of the proposed method against detector attack using first-order SPAM features is almost similar to themselves second-order.

Unlike the Wu *et al.'s* [18] and the Yang *et al.'s* [15] methods, our proposed method resists the RS steganalysis attack, according to Figure 7a. Also, the security of the proposed method against detector attack using first-order and second-order SPAM features is generally better than Khodaei *et al.'s* that is presented in [19].

## 5. CONCLUSION

To evaluate the proposed method, the difference between the cover-image and its stego-image was calculated. In addition, we computed the PSNR value and the embedding capacity using various range divisions. As per the experimental results, the visual quality obtained by our method is higher than its corresponding value achieved by the other related methods, explained in this paper. Furthermore, we hide larger embedding capacity compared with these methods using our new range division. Finally, our proposed method was robust against RS and steganalysis detector attacks.

## REFERENCES

[1] R.J. Anderson, *et al.*, "On the limits of steganography", *IEEE Journal of Selected Areas in Communications*, 1998, vol. 16, pp. 474–481.
[2] D. Artz, *et al.*, "Digital steganography: hiding data within data", *IEEE Internet Computing.*, 2001, vol. 5, pp. 75–80.
[3] A. Cheddad, *et al.*, "Digital image steganography: Survey and analysis of current methods", *Signal Processing*, 2010, vol. 90, pp. 727-752.
[4] A. Nissar, *et al.*, "Classification of steganalysis techniques: A study", *Digital Signal Processing*, 2010, vol. 20, pp. 1758–1770.
[5] J. Hsiao, *et al.*, "An adaptive steganographic method based on the measurement of just noticeable distortion profile", *Image and Vision Computing*, 2011, vol. 29, pp. 155-166.
[6] H. Sajedi, *et al.*, "Using contourlet transform and cover selection for secure steganography", *International Journal of Information Security*, 2010, vol. 9, pp. 337–352.
[7] Y. Xueyi, *et al.*, "A jpeg steganographic Method Based on Syndrome-Trellis Codes", *Journal of Theoretical & Applied Information Technology*, 2013, vol. 47, pp. 194-200.
[8] L. Fan, *et al.*, "An extended matrix encoding algorithm for steganography of high embedding efficiency", *Computers and Electrical Engineering*, 2011, vol. 37, pp. 973–981.
[9] A. Sur, *et al.*, "Secure Steganography Using Randomized Cropping", *Transactions on Data Hiding and Multimedia Security VII*, 2012, vol.7110, pp. 82–95.
[10] W.N. Lie, *et al.*, "*Data hiding in images with adaptive numbers of least significant bits based on the human visual system*", Proceedings of International Conference on Image Processing, 1999, ICIP 99, vol.1, pp. 286–290.
[11] C.-K. Chan, *et al.*, "Hiding data in images by simple LSB substitution", *Pattern Recognition*, 2004, vol. 37, pp. 469–474.
[12] C.-H. Yang, "Inverted pattern approach to improve image quality of information hiding by LSB substitution", *Pattern Recognition.*, 2008, vol. 41, pp. 2674–2683.
[13] X. Li, *et al.*, "A generalization of LSB matching", *IEEE Signal Processing Letters*, 2009, vol. 2, pp. 69–72

[14] D.-C. Wu, *et al.*, "A steganographic method for images by pixel value differencing", *Pattern Recognition Letters*, 2003, vol. 24, pp. 1613–1626
[15] C.-H. Yang, *et al.*, "Adaptive data hiding in edge areas of images with spatial LSB domain systems", *Information Forensics and Security, IEEE Transactions on*, 2008, vol. 3, pp. 488–497.
[16] W. Luo, *et al.*, "Edge Adaptive Image Steganography Based on LSB Matching Revisited", *Information Forensics and Security, IEEE Transactions on*, 2010, vol. 5, pp. 201-214.
[17] N. Jain, *et al.*, "Image Steganography Using LSB and Edge – Detection Technique", *International Journal of Soft Computing and Engineering*, 2012, vol. 2, pp. 217-222.
[18] H.-C. Wu, *et al.*, "*Image steganographic scheme based on pixel-value differencing and LSB replacement methods*", IEE Proceedings Vision, Image and Signal Processing, 2005, vol. 152, pp. 611–615.
[19] M. Khodaei, *et al.*, "New adaptive steganographic method using least significant-bit substitution and pixel-value differencing", *IET Image Processing,* 2012, vol. 6, pp. 677–686.
[20] M. Gadiparthi, *et al.*, "A high capacity steganographic technique based on LSB and PVD modulus methods", *International Journal of Computer Applications*, 2011, vol. 22, pp. 8-11
[21] G. Kaur, *et al.*, "A Steganography Implementation based on LSB & DCT", *International Journal for Science and Emerging, Technologies with Latest Trends*, 2012, vol. 4, ISSN No. (Online):2250-3641, ISSN No. (Print): 2277-8136, pp. 35-41.
[22] J. Fridrich, *et al.*, "Reliable detection of LSB steganography in color and grayscale images", *ACM Workshop on Multimedia and Security*, 2001, pp. 27–30.
[23] T. Pevny, *et al.*, "Steganalysis by subtractive pixel adjacency matrix", *IEEE Transaction Inforation Forensics Security*, 2010, vol. 5, pp. 215–224

## BIOGRAPHIES OF AUTHORS

**Mojtaba Bahmanzadegan Jahromi.** Student of Faculty of Computer and Information Technology Engineering, Qazvin Branch Islamic Azad University, Qazvin, Iran. *Research Interests:* Image Processing, Computer Security, Fuzzy Logic, etc.
*Education:*
Payam Noor *University,* Jahrom, Iran. B.Sc. in Computer Science, 2010,

**Karim Faez.** Professor of EE Department Amirkabir University of Technology, Tehran, Iran
*Research Interests:* Computer Security, Image Processing, Pattern Recognition, etc.
*Education:*
Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran. B.Sc. in Electrical Engineering, 1973,
University of California, Los Angeles. M. Sc. in Computer Science, 1977.
University of California, Los Angeles. Ph. D. in Computer Science, 1980.