

Histogram Shifting as a Data Hiding Technique: An Overview of Recent Developments

Yasaman Zandi Mehran¹, Mona Nafari^{2,*},
Alireza Nafari³, and Nazanin Zandi Mehran⁴

¹ Islamic Azad University, Shahre-e-Rey Branch, Tehran, Iran
zandi@srbiau.ac.ir

² Razi University of Kermanshah,
Department of Electrical Engineering, Kermanshah, Iran
Fax: 0098-21-88614966

m.nafari@razi.ac.ir, Mona_nafari_1362@yahoo.com

³ Amir Kabir University of Technology,
Department of Electrical and Mechanical Engineering, Tehran, Iran
ali.heisenberg@aut.ac.ir

⁴ Amir Kabir University of Technology,
Department of Biomedical Engineering, Tehran, Iran
nazaninznd@aut.ac.ir

Abstract. Histogram shifting is a data hiding technique which has been proposed since 2004. In this paper, we provide an overview of recent contributions pertaining to the Histogram shifting technique. It discusses on this method and its development in terms of payload capacity and image quality. From these discussions, we can state that which schemes are beneficial in terms of capacity-PSNR control. Overallly, Histogram shifting is a valuable technique. Its practical applications are expected to grow in years to come.

Keywords: Histogram shifting, Pseudo code, predictive coding, histogram modification, difference image, block, correlation, sub-sampling.

1 Introduction

Security problems, such as interception, modification, duplication, etc on the Internet, have encountered critical situation [1]. Data hiding conceals the secret data in the cover medium (which can be a digital image) as a method that have been proposed to protect the security [2]. Reversible recovery of the cover image is preferable for some applications such as medical diagnosis, legal documents, etc [3]. Lossless restoration of the original image is required after the message is extracted. Several reversible data hiding schemes have been proposed [4][5][6]. These schemes can be divided to three category: spatial domain, frequency domain, and index domain. In the spatial domain schemes, Celik et al proposed generalized least significant bit(G-LSB) method [7]. Difference expansion data hiding was proposed by Tian in 2003 [8], in which the

* Corresponding author.

redundancy of the pixels is explored. The histogram of the pixels in the image was applied by Ni et al [9] in 2006. The peak and zero pair of the histogram was explored. In the histogram-based data hiding, the number of pixels in the peak point, represents hiding capacity. Another reversible data hiding was proposed by Tsai et al[10] in 2005, in which a pair wise logical computation(PWLC) for data hiding was utilized. In the frequency domain, some reversible data hiding schemes was introduced. Fridrich et al [11] proposed LSB-based data hiding in 2001. The LSB plane of the quantized DCT coefficients was compressed. The compressed data with the secret message were embedded in the LSB bits of the coefficients. In 2002, Xuan et al. [12] explored the relationship between the bit planes of the discrete wavelet transformation (DWT) coefficients. Kamstra and Heijmans [13] proposed a data hiding scheme in 2005 that employed DWT coefficients to embed secret data. In the index domain, secret data is embedded in the vector quantized image. In 2005, a data hiding scheme was proposed by Chang and Wu [14] which was based on vector quantization (VQ).

The rest of this paper is organized as follows: In Section 2, histogram-based data hiding techniques are briefly described. In Section 3, the simulation results are illustrated. Conclusions are made in Section 4.

2 Related Works

In this paper reversible histogram based data hiding techniques and its developments since 2004 has been described, which its basic idea proposed by Ni et al in 2004 [15].

In the basic histogram shifting data hiding, first a zero point and a peak point are found. Zero point (or minimum number of pixels) corresponds to the gray scale value which no pixel in the cover image assumes. Peak point corresponds to the gray scale value with the maximum number of pixels in the cover image. The goal of finding the peak points is to increase the payload capacity as large as possible. Since the number of bits which are embedded into an image equals to the number of pixels associated with the peak points, two or more pairs of zero and peak points can be used for increasing the capacity and generally in any method which is based on image histogram (directly or indirectly), it is aimed to increase the number of peak points. Here for simplicity in illustration of the principle of the algorithm, only one pair of zero-peak values are applied.

Thus, in the next step after finding zero-peak pair, the image is scanned in a sequential order. The gray scale value of pixels between peak and zero points are incremented by 1 unit. This step is equivalent to shifting the gray scale values of pixels between peak and zero points in the histogram, to the right hand side by 1, leaving the gray scale value of peak points empty. The whole image is scanned again, once a pixel with gray scale value of peak point is encountered, if the corresponding bit to be embedded is 1, the pixel is incremented by 1, otherwise the pixel remains intact. It can be observed that the payload capacity of this algorithm equals to the number of pixels that assume the gray scale value of the peak point, when there is only one pair of zero-peak points. Clearly if the required capacity is greater than the actual capacity, more pairs of peak and zero points (maximum and minimum) is needed. The embedding algorithms which is presented below use multiple pairs of maximum and minimum points.

2.1 Pseudo Code Data Hiding Algorithm with Multiple Pairs of Peak and Zero Points

In this section for the case of three pairs of peak and zero points, a pseudo code embedding algorithm is presented which has been proposed by Ni et al in 2006 [16]. It is aimed to generate this code to reach the cases where any number of multiple pairs of peak and zero points are used. This algorithm has two phases like any other data hiding methods: Embedding process, Extraction process. These processes are described as the following.

2.1.1 Embedding Process

First for an $M \times N$ image, with gray scale value of each pixel $x \in [0, 255]$:

Step.1: Generate its histogram $H(x)$.

Step.2: In the histogram $H(x)$, find three minimum point $H(b_1)$, $H(b_2)$, $H(b_3)$.

Assume three minimum points satisfy the following condition:

$$0 < b_1 < b_2 < b_3 < 255$$

Step.3: In the intervals of $(0, b_1)$ and (b_2, b_3) , find the maximum points $h(a_1)$, $h(a_3)$, respectively, and assume $a_1 \in (0, b_1)$, $a_3 \in (b_2, b_3)$.

Step.4: In the intervals (b_1, b_2) and (b_2, b_3) , find the maximum points in each

interval. Assume they are: $h(a_{12})$, $h(a_{21})$, $b_1 < a_{12} < a_{21} < b_2$ and $h(a_{23})$, $h(a_{32})$,

$$b_2 < a_{23} < a_{32} < b_3.$$

Step.5: Find a point having a larger histogram value in each of the following three maximum points pairs $(h(a_1), h(a_{12}))$, $(h(a_{21}), h(a_{23}))$, and $(h(a_{32}), h(a_3))$, respectively.

Assume $h(a_1)$, $h(a_{23})$, $h(a_3)$ are the three selected maximum points.

Step.6: $(h(a_1), h(b_1))$, $(h(a_{23}), h(b_2))$ and $(h(a_3), h(b_3))$ are the three pairs of maximum and minimum points. For each of these three pairs, apply Steps 3-5. All of these three pairs are treated as cases of peak and zero points pairs.

2.1.2 Extraction Process

For simplicity only one pair of peak and zero points is described here, because the general cases of multiple pairs of maximum and minimum points can be decomposed as a one pair case.

Assume the grayscale value of peak and zero points are a and b , respectively. Assume $a < b$ in the marked image of size $M \times N$, where $x \in [0, 255]$.

Step.1: Scan the marked image in the same sequential order as that used in the embedding procedure. If a pixel with its gray scale value $a + 1$ is encountered, a bit "1" is extracted. If a pixel with its value a is encountered, a bit "0" is extracted.

Step.2: Scan the image again, for any pixel with gray scale value $x \in (a, b]$, the pixel value is subtracted by 1.

Step.3: If there is overhead information found in the extracted data, set the pixel grayscale value (whose coordinate (i, j) is saved in the overhead) as b .

2.2 Data Hiding Scheme Using Predictive Coding and Histogram Shifting

The similarity of neighboring pixels in the cover image is applied in this scheme which is proposed by Tsai et al in 2009[17]. By using the prediction technique and the residual histogram of the predicted errors of the host image, the secret data is embedded in the residual image by using a modified histogram-based approach. To increase the hiding capacity of the histogram-based scheme, the linear prediction technique is employed to process the image. Here, the pixels of the image are predicted from the residual image. Secret data are then embedded in the residual image by using a modified histogram-based approach. The proposed scheme consists of two procedures: the embedding procedure and the extraction and image restoration procedure.

2.2.1 The Embedding Procedure

According to the histogram based data hiding, the more the amplitude of a given histogram changes, the more the hiding capacity is. To apply the similarity between neighboring pixels, first the cover image is divided into 3×3 , 5×5 , 7×7 , ... pixel blocks. One pixel in each block is selected as the basic pixel for prediction. All pixels in the block are processed by the linear prediction technique to generate the prediction errors, which is called the residual values. By calculating the difference between the basic pixel and each pixel, the prediction error is determined. Each block is sequentially processed in the same way. After processing all blocks, the residual image is generated. Next, the histogram of the residual image is generated. All values in the residual image are not employed to generate the histogram. After finding the occurrences of residual values that correspond to these non-basic pixels in the cover image, the residual histogram is generated. The residual histogram can be divided into two parts: non-negative histogram(NNH) and negative histogram (NH). After the residual histogram is generated, the secret data are embedded in the residual values of the residual image. If sb is the secret data to be embedded, first the pairs of peak and zero points in NNH and NH have to be searched. If there isn't enough space to embed the secret data, more pairs of peak and zero points in NNH and NH have to be searched.

Each residual value in the peak point is employed to carry 1-bit secret bit sb . One of two possible cases is found for such residual value. If the value of sb equals 1, no change is needed for the residual value. Otherwise, the residual value is shifted closer to the value of the zero point by 1. The remaining residual value ranges between the peak and zero points are shifted closer to the value of the zero point by 1. The residual values that are outside the peak and zero pairs remain unchanged. The modification for secret embedding is employed to both NNH and NH. After the secret data is embedded in the residual image, the residual stego-image is generated. By performing the reverse linear prediction on the residual stego- image, the stego-image of the proposed scheme is obtained. The residual values corresponding to the basic pixels in the cover image are not included in calculating the residual histogram. In other words, residual values are used in each block of pixels in the cover image. In addition, to provide a good image quality for the stego image, the absolute distance between the original residual value and its modified values is at most 1. Therefore, the absolute distance between one pixel in the cover image and its corresponding pixel in the stego image is at most 1.

2.2.2 The Extraction and Restoration Procedure

When the stego image and the pairs of peak and zero points are ready, the procedure can be started. First, the linear prediction technique used in the embedding procedure is applied to the stego image. Then, the residual stego-image is obtained. The pixel of the residual stego image is examined to extract the embedded secret data sb and to recover the original image. This procedure is similar to that of the original histogram based extraction procedure. Two different cases are considered:

1-If pixel of the residual stego image is not within the peak and zero points range, this pixel is skipped and the pixel value remains unchanged for image reconstruction.

2-Otherwise, three cases are discussed:

- a) In the first case, if pixel value is in the peak point, 1-bit secret data valued at 1 is extracted and the value of pixel is unchanged.
- b) In the second case, if the absolute difference between x and the peak point is 1, 1-bit secret data valued at 0 is extracted and the pixel value is replaced by the value of the peak point.
- c) finally, the remaining pixels are shifted close to peak point by 1 and no secret data is extracted. After that, the embedded secret data is extracted from the stego image. The original image can be restored by performing reverse linear prediction to the reconstructed stego image.

2.3 Data Hiding Based on Histogram Modification of Difference Images

In this section, another method of histogram based data hiding technique described which has been proposed by Chia-Chen Lin et al [18] in 2008. In this scheme which is based on the difference image histogram modification, the peak point are used to hide secret messages. This method is divided into three phase: Difference histogram creating, hiding and extracting phases.

2.3.1 Creating the Histogram Phase

A difference image of an image must be generated before the hiding phase, to create a enough free space for data hiding. For a gray scale image $I(i, j)$ with $M \times N$ pixels in size, a difference image $D(i, j)$, with $M \times (N - 1)$ pixels can be generated from the original image $I(i, j)$ by using following formula:

$$D(i, j) = |I(i, j) - I(i, j+1)| \quad \begin{matrix} 0 \leq i \leq M-1 \\ 0 \leq j \leq N-2 \end{matrix} \quad (1)$$

The maximum pixel values in a difference image tend to be around value 0. The peak point in the histogram of a difference image is used to create the free space for hiding messages. Therefore, by using the difference image histogram, a larger number of messages can be hid in comparison with the original image.

2.3.2 Hiding Phase

Step 1: Divide the original cover image into blocks $A \times B$ in size. Generate a difference image $D_b(i, j)$ of size $A \times B - 1$ for each block by using following formula:

$$D_b(i, j) = |I_b(i, j) - I_b(i, j+1)|, 0 \leq i \leq A-1 \quad (2)$$

$$0 \leq j \leq B-2 \quad 0 \leq b \leq \frac{M \times N}{A \times B} - 1$$

Step 2: Generate the histogram of the difference image $D_b(i, j)$ and record the peak point p_b for each block.

Step 3: If the pixel value $D_b(i, j)$ of block b is larger than the peak point p_b of block b , change the pixel value $D_b(i, j)$ of block b to $D_b(i, j)+1$. Otherwise, the pixel value $D_b(i, j)$ remains unchanged. The modification principle is defined as

$$D'_b(i, j) = \begin{cases} D_b(i, j)+1 & \text{if } D_b(i, j) > p_b \\ D_b(i, j) & \text{o.w.} \end{cases} \quad (3)$$

For $0 \leq i \leq A-1, 0 \leq j \leq B-2, 0 \leq b \leq \frac{M \times N}{A \times B} - 1$, where p_b is the peak point of block b .

Step 4: For the modified difference image D'_b , the pixels having grayscale values the same as peak point p_b , can be modified as follows to hide embedded message bit m :

$$D''_b(i, j) = \begin{cases} D'_b(i, j)+m & \text{if } D'_b(i, j) = p_b \\ D'_b(i, j) & \text{o.w.} \end{cases} \quad (4)$$

where p_b is the peak point of block b , and $m \in \{1, 0\}$.

Step 5: Use the original image and its hidden difference image to construct the marked image by performing the inverse transformation T^{-1} . For the first two pixels in each row, the inverse operation is as follows:

$$S_b(i, 0) = \begin{cases} I_b(i, 0)+m & \text{if } I_b(i, 0) > I_b(i, 1) \\ I_b(i, 1)+D''_b(i, 0) & \text{o.w.} \end{cases} \quad (5)$$

$$S_b(i, 1) = \begin{cases} I_b(i, 0)+D''_b(i, 0) & \text{if } I_b(i, 0) < I_b(i, 1) \\ I_b(i, 1) & \text{o.w.} \end{cases} \quad (6)$$

For $0 \leq i \leq A-1, 0 \leq b \leq \frac{M \times N}{A \times B} - 1$.

For any residual pixels, the inverse operation is defined as:

$$S_b(i, j) = \begin{cases} S_b(i, j-1)+D''_b(i, j-1) & \text{if } I_b(i, j-1) < I_b(i, j) \\ S_b(i, j-1)-D''_b(i, j-1) & \text{o.w.} \end{cases} \quad (7)$$

For $0 \leq i \leq A-1, 0 \leq j \leq B-2, 0 \leq b \leq \frac{M \times N}{A \times B} - 1$.

2.3.3 Extracting and Restoring Phase

The embedded message is extracted in this phase in addition to restoration of original image. The basic steps for the extraction and restoration process are as follows.

Step 1: Divide the received marked image into blocks $A \times B$ in size. Generate the difference image $SD_b(i, j)$ of block b from the received marked image by using the following formula:

$$SD_b(i, j) = |S_b(i, j) - S_b(i, j+1)| \quad (8)$$

$$\text{For } 0 \leq i \leq A-1 \quad 0 \leq j \leq B-2 \quad 0 \leq b \leq \frac{M \times N}{A \times B} - 1.$$

Step 2: Perform the embedded message extracting on the difference image $SD_b(i, j)$ of block b by using the following rule:

$$m = \begin{cases} 0 & \text{if } SD_b(i, j) = p_b \\ 1 & \text{if } SD_b(i, j) = p_b + 1 \end{cases} \quad (9)$$

$$\text{For } 0 \leq i \leq A-1 \quad 0 \leq j \leq B-2 \quad 0 \leq b \leq \frac{M \times N}{A \times B} - 1$$

Where p_b is the received peak point of block b . We first scan the entire difference image of block b . For block b , if the pixel with p_b is encountered, bit 0 is extracted. If the pixel with $(p_b + 1)$ is encountered, bit 1 is extracted.

Step 3: Remove the embedded message from the difference image $SD_b(i, j)$ for block b by using the following formula:

$$SD'_b(i, j) = \begin{cases} SD_b(i, j) - 1 & \text{if } SD_b(i, j) = p_b + 1 \\ SD_b(i, j) & \text{o.w.} \end{cases} \quad (10)$$

Step 4: Shift some pixel values in the difference image $SD'_b(i, j)$ to obtain its reconstructed original difference image $RD_b(i, j)$ according to:

$$RD_b(i, j) = \begin{cases} SD'_b(i, j) - 1 & \text{if } SD_b(i, j) > p_b + 1 \\ SD'_b(i, j) & \text{o.w.} \end{cases} \quad (11)$$

$$\text{For } 0 \leq i \leq A-1 \quad 0 \leq j \leq B-2 \quad 0 \leq b \leq \frac{M \times N}{A \times B} - 1$$

Step 5: Finally, obtain the recovered original image $RH_b(i, j)$ by performing the inverse transformation T^{-1} . Similar to Step 5 in the hiding phase, for the first two pixels of each row the inverse operation is expressed as:

$$RH_b(i, 0) = \begin{cases} S_b(i, 0) & \text{if } S_b(i, 0) \leq S_b(i, 1) \\ S_b(i, 1) + RD_b(i, 0) & \text{o.w.} \end{cases} \quad (12)$$

$$RH_b(i, 1) = \begin{cases} S_b(i, 0) + RD_b(i, 0) & \text{if } S_b(i, 0) \leq S_b(i, 1) \\ S_b(i, 1) & \text{o.w.} \end{cases} \quad (13)$$

For the remaining pixels, the corresponding inverse operation is shown as

$$RH_b(i, j) = \begin{cases} RH_b(i, j-1) + RD_b(i, j-1) & \text{if } S_b(i, j-1) \leq S_b(i, j) \\ RH_b(i, j-1) - RD_b(i, j-1) & \text{if } o.w \end{cases} \quad (14)$$

For $0 \leq i \leq A-1$ $0 \leq j \leq B-2$ $0 \leq b \leq \frac{M \times N}{A \times B} - 1$.

Because the hiding algorithm is based on a multilevel concept, the algorithm can be performed repeatedly to convey a large amount of embedded messages.

2.4 Data Hiding Scheme Based on Three-Pixel Block Differences

In this section a data hiding scheme has been proposed that embeds a message into an image using the two differences, between the first and the second pixels as well as between the second and the third pixels in a three-pixel block. In this method, the term “histogram” is not used necessarily, but its fundamental concepts have been indirectly applied, which are using peak points, zero points and shifting the gray scale between these two values. This scheme has been proposed by Ching-Chiuan Lin et al in 2008 [19]. In the cover image, an absolute difference between a pair of pixels is selected to embed the message. In the best case, a three-pixel block can embed two bits and only the central pixel needs to be increased or decreased by 1. First an image is divided into non-overlapping three-pixel blocks, where the maximum and minimum allowable pixel values are 255 and 0, respectively.

2.4.1 Embedding Process

Let $g(d)$ be the number of pixel pairs with absolute difference equal to d , where $0 \leq d \leq 253$ and pixel pairs in the block which contains a pixel value equal to 0 or 255 are not considered when calculating $g(d)$. Before embedding a message, the proposed scheme selects a pair of differences M and m such that $g(M) \geq g(M')$ and $g(m) \leq g(m')$ for all $0 \leq M', m' \leq 253$. Let (b_{i0}, b_{i1}, b_{i2}) denote a block i with pixel values equal to b_{i0}, b_{i1} , and b_{i2} , and $\max(b_{i0}, b_{i1}, b_{i2})$ and $\min(b_{i0}, b_{i1}, b_{i2})$ denote the maximum and minimum pixel values in the block, respectively. First, blocks satisfying the following two conditions are selected:

- (a) $1 \leq b_{i0}, b_{i1}, b_{i2} \leq 254$
- (b) $\max(b_{i0}, b_{i1}, b_{i2}) = 254$ and $\min(b_{i0}, b_{i1}, b_{i2}) = 1$.

For each block i satisfying the above conditions (a) and (b), call the embedding procedure shown in Figure.1.to embed the message. The brief procedure is described in two steps. The steps are expanded in detail. After invoking the embedding procedure, if $\max(b_{i0}, b_{i1}, b_{i2}) = 255$ or $\min(b_{i0}, b_{i1}, b_{i2}) = 0$, record block i in the overhead information. For each selected block i , the sender performs the following actions:

Step1: Increase d_{i0} by 1 if $M+1 \leq d_{i0} \leq m-1$, and increase d_{i1} by 1 if $M+1 \leq d_{i1} \leq m-1$, where $d_{i0} = |b_{i0} - b_{i1}|$ and $d_{i1} = |b_{i1} - b_{i2}|$;

Step2: Embed a message into block i if $d_{i0} = M$ or $d_{i1} = M$;

Then, the sender scans the image again and performs the following actions for each block i with $2 \leq b_{i0}, b_{i1}, b_{i2} \leq 253$.

Step 1': Increase d_{i0} by 1 if $M+1 \leq d_{i0} \leq m-1$, and increase d_{i1} by 1 if $M+1 \leq d_{i1} \leq m-1$.

Step 2': Embed the overhead information and the residual message into block i if $d_{i0} = M$ or $d_{i1} = M$.

```

Procedure of embedding:
if  $d_{i0} == M$  {
    if  $d_{i1} == M$ 
        embed 2 bits;
    else {
        if  $M < d_{i1} < m$ 
            embed 1 bit and increase difference;
        else
            embed 1 bit and leave unchanged;
    }
}
Else if  $M < d_{i0} < m$  {
    if  $d_{i1} == M$ 
        increase difference and embed 1 bit;
    else {
        if  $M < d_{i1} < m$ 
            increase 2 differences;
        else
            increase difference and leave unchanged;
    }
}
else {
    if  $d_{i1} == M$ 
        leave unchanged and embed 1 bit;
    else {
        if  $M < d_{i1} < m$ 
            leave unchanged and increase difference;
        else
            do nothing;
    }
}

```

Fig. 1. Embedding procedure

Then go to step 1 until the message is completely embedded.

2.4.2 Extraction Process

For extraction process scan the stego-image block by block in the order the message was embedded. For each block i with $1 \leq b_{i0}, b_{i1}, b_{i2} \leq 254$, perform the actions in accordance with the conditions listed in Figure 2. After the respective actions have been completed, save the extracted message bits in the list 1 if $\min(b_{i0}, b_{i1}, b_{i2}) = 1$ or $\max(b_{i0}, b_{i1}, b_{i2}) = 254$, and save the extracted message bits in the list 2, if $2 \leq b_{i0}, b_{i1}, b_{i2} \leq 253$. List-1 contains a part of the message embedded in step 1,2 and List-2 contains the message embedded in step 1', 2'.

Item Conditions	Actions
1 $d_{i0} == M$ and $d_{i1} == M$	Extract "00"
2 $d_{i0} == M$ and $d_{i1} == M + 1$ and $b_{i1} > b_{i2}$	Extract "01", $b_{i2} = b_{i2} + 1$
3 $d_{i0} == M$ and $d_{i1} == M + 1$ and $b_{i1} < b_{i2}$	Extract "01", $b_{i2} = b_{i2} - 1$
4 $d_{i0} == M$ and $M + 1 < d_{i1}$ and $b_{i1} > b_{i2}$	Extract "0", $b_{i2} = b_{i2} + 1$
5 $d_{i0} == M$ and $M + 1 < d_{i1}$ and $b_{i1} < b_{i2}$	Extract "0", $b_{i2} = b_{i2} - 1$
6 $d_{i0} == M$ and $(d_{i1} < M \text{ or } d_{i1} > m)$	Extract "0"
7 $d_{i0} == M + 1$ and $d_{i1} == M$ and $b_{i0} < b_{i1}$	Extract "10", $b_{i0} = b_{i0} + 1$
8 $d_{i0} == M + 1$ and $d_{i1} == M$ and $b_{i0} > b_{i1}$	Extract "10", $b_{i0} = b_{i0} - 1$
9 $d_{i0} == M + 1$ and $d_{i1} == M + 1$ and $b_{i0} < b_{i1} < b_{i2}$	Extract "11", $b_{i0} = b_{i0} + 1$, $b_{i2} = b_{i2} - 1$
10 $d_{i0} == M + 1$ and $d_{i1} == M + 1$ and $b_{i0} > b_{i1} > b_{i2}$	Extract "11", $b_{i0} = b_{i0} - 1$, $b_{i2} = b_{i2} + 1$
11 $d_{i0} == M + 1$ and $d_{i1} == M + 1$ and $b_{i0} < b_{i1} > b_{i2}$	Extract "11", $b_{i1} = b_{i1} - 1$
12 $d_{i0} == M + 1$ and $d_{i1} == M + 1$ and $b_{i0} > b_{i1} < b_{i2}$	Extract "11", $b_{i1} = b_{i1} + 1$
13 $d_{i0} == M + 1$ and $M + 1 < d_{i1}$ and $b_{i0} > b_{i1} > b_{i2}$	Extract "1", $b_{i0} = b_{i0} - 1$, $b_{i2} = b_{i2} + 1$
14 $d_{i0} == M + 1$ and $M + 1 < d_{i1}$ and $b_{i0} < b_{i1} < b_{i2}$	Extract "1", $b_{i0} = b_{i0} + 1$, $b_{i2} = b_{i2} - 1$
15 $d_{i0} == M + 1$ and $M + 1 < d_{i1}$ and $b_{i0} < b_{i1} > b_{i2}$	Extract "1", $b_{i1} = b_{i1} - 1$
16 $d_{i0} == M + 1$ and $M + 1 < d_{i1}$ and $b_{i0} > b_{i1} < b_{i2}$	Extract "1", $b_{i1} = b_{i1} + 1$
17 $d_{i0} == M + 1$ and $(d_{i1} < M \text{ or } d_{i1} > m)$ and $b_{i0} < b_{i1}$	Extract "1", $b_{i0} = b_{i0} + 1$
18 $d_{i0} == M + 1$ and $(d_{i1} < M \text{ or } d_{i1} > m)$ and $b_{i0} > b_{i1}$	Extract "1", $b_{i0} = b_{i0} - 1$
19 $M + 1 < d_{i0}$ and $d_{i1} == M$ and $b_{i0} > b_{i1}$	Extract "0", $b_{i0} = b_{i0} - 1$
20 $M + 1 < d_{i0}$ and $d_{i1} == M$ and $b_{i0} < b_{i1}$	Extract "0", $b_{i0} = b_{i0} + 1$
21 $M + 1 < d_{i0}$ and $d_{i1} == M + 1$ and $b_{i0} > b_{i1} > b_{i2}$	Extract "1", $b_{i0} = b_{i0} - 1$, $b_{i2} = b_{i2} + 1$
22 $M + 1 < d_{i0}$ and $d_{i1} == M + 1$ and $b_{i0} < b_{i1} < b_{i2}$	Extract "1", $b_{i0} = b_{i0} + 1$, $b_{i2} = b_{i2} - 1$
23 $M + 1 < d_{i0}$ and $d_{i1} == M + 1$ and $b_{i0} < b_{i1} > b_{i2}$	Extract "1", $b_{i1} = b_{i1} - 1$
24 $M + 1 < d_{i0}$ and $d_{i1} == M + 1$ and $b_{i0} > b_{i1} < b_{i2}$	Extract "1", $b_{i1} = b_{i1} + 1$
25 $M + 1 < d_{i0}$ and $M + 1 < d_{i1}$ and $b_{i0} > b_{i1} > b_{i2}$	$b_{i0} = b_{i0} - 1$, $b_{i2} = b_{i2} + 1$
26 $M + 1 < d_{i0}$ and $M + 1 < d_{i1}$ and $b_{i0} < b_{i1} < b_{i2}$	$b_{i0} = b_{i0} + 1$, $b_{i2} = b_{i2} - 1$
27 $M + 1 < d_{i0}$ and $M + 1 < d_{i1}$ and $b_{i0} < b_{i1} > b_{i2}$	$b_{i1} = b_{i1} - 1$
28 $M + 1 < d_{i0}$ and $M + 1 < d_{i1}$ and $b_{i0} > b_{i1} < b_{i2}$	$b_{i1} = b_{i1} + 1$
29 $M + 1 < d_{i0}$ and $(d_{i1} < M \text{ or } d_{i1} > m)$ and $b_{i0} < b_{i1}$	$b_{i0} = b_{i0} + 1$
30 $M + 1 < d_{i0}$ and $(d_{i1} < M \text{ or } d_{i1} > m)$ and $b_{i0} > b_{i1}$	$b_{i0} = b_{i0} - 1$
31 $(d_{i0} < M \text{ or } d_{i0} > m)$ and $d_{i1} == M$	Extract "0"
32 $(d_{i0} < M \text{ or } d_{i0} > m)$ and $d_{i1} == M + 1$ and $b_{i1} < b_{i2}$	Extract "1", $b_{i2} = b_{i2} - 1$
33 $(d_{i0} < M \text{ or } d_{i0} > m)$ and $d_{i1} == M + 1$ and $b_{i1} > b_{i2}$	Extract "1", $b_{i2} = b_{i2} + 1$
34 $(d_{i0} < M \text{ or } d_{i0} > m)$ and $M + 1 < d_{i1}$ and $b_{i1} < b_{i2}$	$b_{i2} = b_{i2} - 1$
35 $(d_{i0} < M \text{ or } d_{i0} > m)$ and $M + 1 < d_{i1}$ and $b_{i1} > b_{i2}$	$b_{i2} = b_{i2} + 1$
36 $(d_{i0} < M \text{ or } d_{i0} > m)$ and $(d_{i1} < M \text{ or } d_{i1} > m)$	Do nothing

Fig. 2. Conditions and their actions for extracting process of data hiding scheme based on three-pixel block differences

2.5 Data Hiding Based on Correlation between Sub-sampled Images

In this scheme, a data hiding method is described which has been proposed by Kim[20] that modifies the difference histogram between sub-sampled images. It applies the spatial correlation in neighboring pixels to achieve higher capacity than other histogram based methods. After explanation of embedding process, extraction process have been described.

2.5.1 Embedding Process

Data embedding procedure, is composed of six steps as follows:

Step 1: Generate sub-sampled versions S_k using two sampling factors $(\Delta u, \Delta v)$ and Eq.(1) by performing sub-sampling from an original image I :

$$S_k(i, j) = I\left(i \cdot \Delta v + \text{floor}\left(\frac{k-1}{\Delta u}\right), j \cdot \Delta u + ((k-1) \bmod \Delta u)\right) \quad (15)$$

Step 2: Determine a reference sub-sampled image S_{ref} to maximize spatial correlation between the sub-sampled images. We select S_{ref} from the following Eq.(16). For example, when $\Delta u = 3, \Delta v = 3$, S_5 is determined as the reference one. It is defined as

$$S_{ref} = \left\lceil \text{Round}\left(\frac{\Delta u}{2} - 1\right) \right\rceil \times \Delta v + \text{Round}\left(\frac{\Delta v}{2}\right) \quad (16)$$

Step 3: Create difference images between the reference and the other destination sub-sampled images denoted by:

$$D_{ref-Des}(k_1, k_2) = S_{ref}(k_1, k_2) - D_{Des}(k_1, k_2) \quad (17)$$

where $0 \leq k_1 \leq M/\Delta v - 1, 0 \leq k_2 \leq N/\Delta u - 1$.

Step 4: Prepare empty bins in each histogram of the difference images according to an embedding level L , where $H = -255, \dots, 255$. Depending on the desired degree, L affects the performance of capacity and the perceptual quality. In order to achieve this, the negative differences and then non-negative differences in the outer regions of a selected embedding range should be shifted left and right, respectively. When shifting H , only the pixels in the destination sub-sampled image are modified. The embedding procedure will use the range $[-L, L]$ in the shifted histogram. The shifted histogram H_s can be calculated as follows:

$$H_s = \begin{cases} H + L + 1 & \text{if } H \geq L + 1 \\ H - L - 1 & \text{if } H \leq -L - 1 \end{cases} \quad (18)$$

Also, this can be obtained by:

$$D'_{ref-Des}(k_1, k_2) = S_{ref}(k_1, k_2) - S'_{Des}(k_1, k_2) \quad (19)$$

where

$$S'_{Des}(k_1, k_2) = \begin{cases} S_{Des}(k_1, k_2) - (L + 1) & \text{if } H \geq L + 1 \\ S_{Des}(k_1, k_2) + (L + 1) & \text{if } H \leq -L - 1 \end{cases} \quad (20)$$

Step 5: Embed message $w(n)$ by modifying H_s , where $w(n) = \{1, 0\}$. The modified difference image D' is scanned. Once a pixel with the difference value of $-L$ or $+L$ is encountered, the message to be embedded is checked. This process is repeated until there are no pixels with the difference value of $\pm L$. Then the embedding level decreases by 1. These scanning and embedding steps are executed until $L < 0$. Likewise, only the pixels in the destination sub-sampled image are modified. The message embedding can be formulated as follows:

$$D''_{ref-Des}(k_1, k_2) = S_{ref}(k_1, k_2) - S''_{Des}(k_1, k_2) \quad (21)$$

Where

$$S''_{Des}(k_1, k_2) = \begin{cases} S'_{Des}(k_1, k_2) + (L + 1) & \text{if } D' = -L, w(n) = 1 \\ S'_{Des}(k_1, k_2) - (L + 1) & \text{if } D' = L, w(n) = 1 \\ S'_{Des}(k_1, k_2) + L & \text{if } D' = -L, w(n) = 0 \\ S'_{Des}(k_1, k_2) - L & \text{if } D' = L, w(n) = 0 \end{cases} \quad (22)$$

$$\text{For } L = 0 \text{ \& } L > 0 \quad (23)$$

Step.6: Finally, obtain the marked image I_w through the inverse of the sub-sampling with the unmodified reference sub-sampled image $S_{ref}(k_1, k_2)$ and the modified destination sub-sampled images $S''_{Des}(k_1, k_2)$

2.5.2 Extraction and Restoration Algorithm

The extraction and recovery steps are as follows:

Step 1: Obtain two sampling factors $(\Delta u, \Delta v)$ and the embedding level L from the *LSB* of the selected pixels using the secret key.

Step 2: Generate sub-sampled versions in Eq.(1) by performing sub-sampling from the marked image I_w using the sampling factors in Step1.

Step 3: Determine a reference sub-sampled image S_{ref} by Eq.(2).

Step 4: Create difference images between the unmarked reference sub-sampled image S_{ref} and the other marked destination sub-sampled images denoted by:

$$D_{ref-w}(k_1, k_2) = S_{ref}(k_1, k_2) - S_w(k_1, k_2) \quad (24)$$

where $0 \leq k_1 \leq M/\Delta v - 1, 0 \leq k_2 \leq N/\Delta u - 1$ and $w = 1, \dots, \Delta u \times \Delta v$

Step 5: Extract the hidden message $w(n)$ from each difference image. The extraction process is the inverse of embedding process. After a new variable L' is set to 0, the difference image D is scanned. Once a pixel with the difference value of 1 is encountered, bit 1 is restored. If the pixel with the difference value of 0 encountered, bit 0 is restored. This process is repeated until there are no pixels with the difference value of 0 and 1. After L' is increased by 1, the difference image is scanned again, and the message is extracted using the following rule:

$$w(n) = \begin{cases} 0 & \text{if } D = 2L' \text{ or } -2L' \\ 1 & \text{if } D = 2L' + 1 \text{ or } -(2L' + 1) \end{cases} \quad (25)$$

This scanning and extracting process is executed until $L' > L$.

Step 6: Remove the hidden message $w(n)$ from the difference images.

$$w(n) = \begin{cases} 0 & \text{if } D = 2L' \text{ or } -2L' \\ 1 & \text{if } D = 2L' + 1 \text{ or } -(2L' + 1) \end{cases} \quad (25)$$

This scanning and extracting process is executed until $L' > L$.

Step 6: Remove the hidden message $w(n)$ from the difference images.

For $L' = 0$

$$S'_w(k_1, k_2) = \begin{cases} S_w(k_1, k_2) + L' & \text{if } D(k_1, k_2) = 2L' \\ S_w(k_1, k_2) + (L' + 1) & \text{if } D(k_1, k_2) = 2L' + 1 \\ S_w(k_1, k_2) + L' & \text{if } D(k_1, k_2) = -2L' \\ S_w(k_1, k_2) - (L' + 1) & \text{if } D(k_1, k_2) = -2L' - 1 \end{cases} \quad (28)$$

For $1 \leq L' \leq L$.

Step 7: Shift each histogram H_s of the difference image back to obtain its original difference histogram H as follows:

$$H = \begin{cases} H_s - L - 1 & \text{if } H_s \geq 2L + 2 \\ H_s + L + 1 & \text{if } H_s \leq -2L - 2 \end{cases} \quad (29)$$

This can be obtained by

$$D_{ref-W}''(k_1, k_2) = S_{ref}(k_1, k_2) - S_W''(k_1, k_2) \quad (30)$$

Where

$$S_W''(k_1, k_2) = \begin{cases} S_W'(k_1, k_2) + (L+1) & \text{if } H_s \geq 2L+2 \\ S_W'(k_1, k_2) - (L+1) & \text{if } H_s \leq -2L-2 \end{cases} \quad (31)$$

Step 8: Finally, obtain the covered original image I through the inverse of the sub-sampling with the sub-sampled images.

In the next section, all of the described methods are simulated to illustrate a comparison in terms of quality and capacity.

3 Simulation Results

Simulation are performed to evaluate the performance of all histogram based data hiding schemes. In terms of embedding capacity and invisibility, the performance of these algorithms are measured by comparing with each other and with Ni et al's scheme. The capacity (bit per pixel) measures the amount of data that can be hidden. The peak signal to noise ratio (PSNR) is utilized to show the distortion or invisibility of the stego-image. For an $M \times N$ gray scale image, the PSNR value is defined as follows:

$$PSNR = 10 \times \log \left(\frac{255^2 \times M \times N}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - I(i, j)')^2} \right) dB \quad (32)$$

where $I(i, j)$ and $I(i, j)'$ denote the pixel values in i th row and j th column of the cover image and the stego-image, respectively. Table 1 & 2 show the maximum payload capacity, in bpp and bits respectively, that the test images can offer using all the schemes which are proposed in sections 2.1. to 2.5. In all simulation, three gray scale images of 512×512 pixels are tested as depicted in Figure.4. The message bits to be embedded were randomly generated using the MATLAB function.

Embedding variables of Kim's scheme[19], $(\Delta u, \Delta v)$ and L is adjusted to 3 and 0, respectively. Kim exploited the fact that the difference values having small magnitudes occurred frequently because of the high spatial correlation between sub-sampled images. The embedding capacity of the Kim's algorithm depends on how many the difference images are used and how many pixels having the difference values between $-L$ and L in each difference image exist. In addition, sampling factors affect embedding capacity. Pixel redundancy and spatial correlation between the determined reference sub-sampled image and the other ones are high at the selected sampling factors. From this result, the performance of capacity versus distortion depends on the characteristics of the images. In Tsai et al.'s scheme [16] the negative histogram and non-negative histogram of the residue image are employed

and provided high capacity enhancement compared to original image histogram. In the simulations, these test images are first divided into blocks of 3×3 pixels and then the linear prediction is performed to generate the residual images. In other words, 8 residual values are generated for each 3×3 block. To evaluate the performance of these methods, the hiding capacity and the stego-image quality are computed. The size of the embedded secret data is determined according to the capacity of the supported image. From these results, the performance of capacity versus distortion depends on the characteristics of the images. Tables 1 and 2 summarize comparison results of histogram-shifting based algorithms for three test images: *Lena*, *Baboon* and *Boat*. The hiding capacity of Chia Chen Lin et al.'s scheme [18] is equal to the sum of the number of pixels associated with the peak points of the blocks in the difference image. Based on the nature of an image, the gray scale values close to 0 in its difference image may be the maximum number of pixels. In addition, the number of pixels that correspond to the peak point in a difference image is always larger than the number in its original image. Based on this property of the difference image histogram, a large amount of embedded messages can be hidden in a marked image in comparison with its original image. The embedding performance of the Chia Chen Lin et al.'s scheme did not achieve more than 0.22 bpp for the test images, because the amount of the peak information for all blocks exceeded the whole embedding capacity. In other words, it suffered from the lack of capacity control due to the need for embedding all peak information for blocks. Experimental results supported that Kim's algorithm achieved high embedding capacity in comparison to other reversible schemes while maintaining the distortion at a low level.

In Ni et al.'s scheme [14] two pairs of maximum and minimum points are utilized in data embedding and extraction process. The histogram-based Ni et al.'s scheme showed the fixed PSNR quality, 48.2dB, but the achievable capacity was a little varied for each image. In Ching-Chiuan Lin scheme [19] the peak signal to noise ratio (PSNR) is utilized to show the distortion of the stego-image. The payload capacities and PSNR values of this method are higher than Ni et al.'s in embedding process. Table 1 shows that in each level Ching Lin's scheme provides higher capacity than other schemes. Because it uses three pixel blocks and two differences in each block. The secret bits are embedded in these differences. Thus it provides high payload capacity. The second scheme that provides higher capacity is the Kim's one that uses the correlation of subsampled image. Since the secret bits are embedded in difference image of each subsampled image with the reference one. Thus it provides a large amount of space for hiding secret bits. Tsai [16] and Chia Lin have similar capacity and Ni et al. scheme have almost 0.25bpp for payload capacity. For comparison of image quality and invisibility, Tsai has the highest PSNR, but the image quality of Ching Lin's scheme is satisfactory. Overall comparison between the existing reversible data hiding techniques in terms of pure payload and the PSNR is presented in Table 1. In terms of embedding capacity (bit) and image quality (dB), the algorithms are compared with each other in 16 levels for the *Lena* as shown in Fig. 4. Ching Lin scheme showed relatively high embedding capacity versus PSNR value, whereas the Tsai scheme had low embedding capacity compared to other schemes.

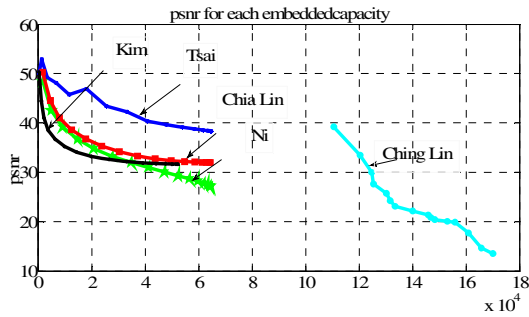


Fig. 3. Comparison of embedding capacity (bit) versus image quality (dB) of methods for test image *lena*

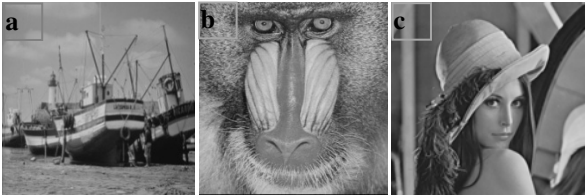


Fig. 4. Test images: (a) Boat; (b) Baboon; (c) Lena

Table 1. Comparison between histogram based data hiding algorithm in terms of the payload capacities and the PSNR values for Lena in 4 level

Level		1	2	3	4
Kim	PSNR(db)	49	43.5	41.5	38.5
	Capacity (Bit/pixel)	0.07	0.225	0.34	0.44
Ni (2006)	PSNR(db)	48.3	48.3	48.2	48.2
	Capacity (Bit/pixel)	0.042	0.14	0.13	0.24
Tsai	PSNR(db)	59	55	52	50
	Capacity (Bit/pixel)	0.02	0.05	0.08	0.21
Ching Lin	PSNR(db)	48.67	43.02	39.64	37.21
	Capacity (Bit/pixel)	0.216	0.38	0.53	0.66
Chia Lin	PSNR(db)	47	43	38	37
	Capacity (Bit/pixel)	0.084	0.087	0.107	0.22

Table 2. The performance of the histogram based data hiding schemes

Methods	Kim's scheme	Ni 2006's scheme	Tsai's scheme	Ching Lin's scheme	Chia Lin's scheme
PSNR(db)Lena	48.9	48.2	50.59	30.0	48.67
Capacity (Bits)	20121	5460	52322	308474	65349
PSNR(db)Baboon	48.7	48.2	51.03	30.4	48.67
Capacity (Bits)	6499	5421	18410	161118	38465
PSNR(db)Boat	68.9	48.2	47.66	30.4	48.67
Capacity (Bits)	21442	7301	53510	307193	56713

4 Conclusions

In this paper, reversible histogram-based data hiding schemes is presented which all of them have been developed in few recent years. All the schemes intend to improve the histogram-based data hiding scheme which embeds secret data in to the peak points of the image histogram. To achieve a higher hiding capacity, more peak-zero pairs are explored instead of using only one pair in each histogram. Experimental results supported that Ching Lin's scheme and Kim's algorithm achieved higher embedding capacity than other reversible schemes while maintaining the distortion at a low level and it has satisfactory PSNR for image quality. The performance of Kim's algorithm can be enhanced by deciding optimum sampling factors considering the characteristics of a given image.

Acknowledgment

The authors gratefully acknowledge the financial and support of this research, provided by the Islamic Azad University, Shahr-e-Rey Branch, Tehran, Iran.

References

1. Cheng, Q., Huang, T.S.: An Additive Approach to Transform-Domain Information Hiding and Optimum Detection Structure. *IEEE Transactions on Multimedia* 3(3), 273–284 (2001)
2. Artz, D.: Digital Steganography: Hiding Data Within Data. *IEEE Internet Computing* 5(3), 75–80 (2001)
3. Podilchuk, C.I., Delp, E.J.: Digital Watermarking: Algorithms and Applications. *IEEE Signal Processing Magazine* 18(4), 33–46 (2001)
4. Wang, R.Z., Lin, C.F., Lin, J.C.: Image Hiding by Optimal LSB Substitution and Genetic Algorithm. *Pattern Recognition* 34(3), 671–683 (2001)
5. Jo, M., Kim, H.D.: A Digital Image Watermarking Scheme Based on Vvector Quantization. *IEICE Transactions on Information and Systems* 9(3), 1054–1105 (2002)
6. Chang, C.C., Tai, W.L., Lin, C.C.: A Reversible Data Hiding Scheme Based on Side-Match Vector Quantization. *IEEE Transactions on Circuits and Systems for Video Technology* 16(10), 1301–1308 (2006)

7. Celik, M.U., Sharma, G., Tekalp, A.M.: Reversible Data Hiding. In: Proceedings of IEEE International Conference on Image Processing, Rochester, NY, vol. 158, pp. 157–160 (2002)
8. Tian, J.: Reversible Data Embedding Using a Difference Expansion. *IEEE Transactions on Circuits and Systems for Video Technology* 13(8), 890–899 (2003)
9. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible Data Hiding. *IEEE Transactions on Circuits and Systems for Video Technology* 16(3), 354–361 (2006)
10. Tai, C.L., Chiang, H.F., Fan, K.C., Chung, C.D.: Reversible Data Hiding and Lossless Reconstruction of Binary Images Using Pair-Wise Logical Computation Mechanism. *Pattern Recognition* 38(11), 1993–2006 (2005)
11. Fridrich, J., Goljan, M., Du, R.: Invertible Authentication. In: Proceedings of SPIE Security Watermarking Multimedia Contents, San Jose, CA, pp. 197–208 (January 2001)
12. Xuan, G., Zhu, J., Chen, J., Shi, Y.Q., Ni, Z., Su, W.: Distortionless Data Hiding Based on Integer Wavelet Transform. *Electronics Letters* 38(25), 1646–1648 (2002)
13. Kamstra, L., Heijmans, H.J.A.M.: Reversible Data Embedding into Images Using Wavelet Techniques and Sorting. *IEEE Transactions on Image Processing* 4(12), 2082–2090 (2005)
14. Chang, C.-C., Wu, W.-C.: A Reversible Information Hiding Scheme Based on Vector Quantization. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3683, pp. 1101–1107. Springer, Heidelberg (2005)
15. Shi, Y.Q., Ni, Z., Zou, D., Liang, C., Xuan, G.: Lossless Data Hiding: Fundamentals, Algorithms and Applications. In: Proc. IEEE Int. Symp. Circuits Syst., Vancouver, BC, Canada, vol. II, pp. 33–36 (May 2004)
16. Ni, Z., Shi, Y.-Q., Ansari, N., Su, W.: Reversible Data Hiding. *IEEE Transactions on Circuit and Systems for Video Technology* 16(3) (March 2006)
17. Tsai, P., Hu, Y.-C., Yeh, H.-L.: Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting. *Signal Processing* 89, 1129–1143 (2009)
18. Lin, C.-C., Tai, W.-L., Chang, C.-C.: Multilevel Reversible Data Hiding Based on Histogram Modification of Difference Images. *Pattern Recognition* 41, 3582–3591 (2008)
19. Lin, C.-C., Hsueh, N.-L.: A Lossless Ddata Hiding Scheme Based on Three-Pixel Block Differences. *Pattern Recognition* 41, 1415–1425 (2008)
20. Kim, K.-S., Lee, M.-J., Lee, H.-Y., Leea, H.-K.: Reversible Data Hiding Exploiting Spatial Correlation Between Sub-sampled Images. *Pattern Recognition* 42, 3083–3096 (2009)