Journal of Software Engineering and Applications, 2017, 10, 56-77
http://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Block-Based Steganographic Algorithm Using Modulus Function and Pixel-Value Differencing

Ahlam K. Al-Dhamari[1], Khalid A. Darabkh[2]

[1]Computer Engineering Department, Hodeidah University, Hodeidah, Yemen
[2]Computer Engineering Deartment, The University of Jordan, Amman, Jordan
Email: ahl_kal@yahoo.com, k.darabkeh@ju.edu.jo

How to cite this paper: Al-Dhamari, A.K. and Darabkh, K.A. (2017) Block-Based Steganographic Algorithm Using Modulus Function and Pixel-Value Differencing. *Journal of Software Engineering and Applications*, **10**, 56-77.
http://dx.doi.org/10.4236/jsea.2017.101004

Received: December 19, 2016
Accepted: January 20, 2017
Published: January 24, 2017

Abstract

The main purpose in developing the steganographic algorithms lies in achieving most of the steganographic objectives which comprise the embedding capacity, imperceptibility, security, robustness and complexity. In this paper, we propose a high quality steganographic algorithm using new block structure which makes a good use of both modulus function and pixel-value differencing, namely, MF-PVD. We have made many experiments with various test images from several galleries, such as USC-SIPI and UWATERLOO-LINK. The performance of our proposed algorithm is verified using three different performance metrics which include peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and embedding capacity (EC). Experimental results and comparisons with six pertinent state-of-art algorithms are given to prove the validation and efficiency of the proposed algorithm.

Keywords

Data Hiding, Steganography, Watermarking, Pixel-Value Differencing, Modulus Function, Performance Metrics

1. Introduction

Nowadays, digital communications become an essential part of our daily life [1] [2] [3] [4] [5]. A lot of applications are based on the Internet and in some cases it is required that communication have to be secret. The wide proliferation of the Internet and wireless networks makes delivery and exchange the digital data easy [6]-[13]. Apart from the problems of the wireless networks which include low bandwidth, insecure links, and high error rate [14]-[22]. Security is an important matter when exchange data over the internet and wireless networks [23]-[29].

In security systems, data hiding is a common discipline to protect data. Data hiding, which concerned by concealing a message (a sequence of bits) in a digital

object or cover, has two important sub-disciplines: steganography and watermarking [30] [31]. Steganography and watermarking are very close to each other and may be correspond but with different requirements and objectives as well, and thus leads to different technical solutions. Watermarking is a technique with a trade-off between the robustness, embedding capacity, and visual quality. Its aim is to obstruct piracy or to prove the proprietorship by imperceptibly modifying the digital media cover. The goal of steganographic techniques is to modify the cover object in imperceptible way, that is, nobody except the intended recipient can be able to identify the modified cover object [32].

The steganography term is not new concept [33]. It is believed that steganography was found during the Golden Age in Greece and literally means "concealed/covert writing" which is derived from two Greek words "Stegan Graphien" [34] [35] [36]. Ancient Greek history mentioned that how ancient Greeks dig secret messages into the waxed wooden tablets and then the melting wax was reapplied to the wood, giving the appearance of a new and unused tablet. Thus, the resulted tablets could be transported without anyone doubting about the presences of a secret message under the wax [37]. An alternate smart method was to shave head of a messenger and tattoo a message or an image on the head of messenger. After the hair grew again, the message would not be discovered till the head was shaved again [38]. At the present time, steganographic algorithms obscure secret data as noise in a cover object that is assumed to be harmless [39].

Generally, the algorithms used in steganography are based on the fact that the modifications occurred in texts, images, audio files and video files are not detected by human visual or auditory systems [40]. There are several steganographic algorithms that have been proposed for data hiding when the cover media is digital images [41] [42] [43]. Since the digital images have a lot of redundant data, there has been an increased interest in using digital images as cover media for steganographic purposes [44] [45]. On the other hand, millions of digital images transfer through the internet each second, therefore it is necessary to say that digital image steganography becomes an important topic in IT security field [46] [47]. However, the remainder of the paper is organized as follows. In Section 2, a literature review of the most related works in the field is presented. Section 3, detailed extensively the proposed algorithm. Experimental results and discussions are provided in Section 4. Section 5 concludes the work and provides future possible directions.

2. Related Work

The first pixel-value differencing (PVD) steganographic algorithm was introduced by [41]. The embedding algorithm of Wu and Tsai's method uses a cover image I sized by $M \times N$. B_i is a sub-block of I, which has two consecutive pixels p_i and p_{i+1} broken down by partitioning I in raster scan order such that $I = \{B_i | i = 1, 2, \cdots, (M \times N)/2\}$. The difference value d_i between p_i and p_{i+1} can be derived by using the following equation:

$$d_i = p_{i+1} - p_i \tag{1}$$

On the other hand, in PVD method a range table has been designed with n contiguous sub-ranges R_j (*i.e.* $R = \{R_j \mid j = 1, 2 \cdots n\}$. The main job of the range table is to provide the lower and upper bound values for each $|d_i|$ that follow each B_i. The lower and upper bound values are called by l_j and u_j, so that we have $R_j \in [l_j, u_j]$. Each sub-range R_j has width W_j which is selected to be a power of 2 and can be calculated by using this equation:

$$W_j = u_j - l_j + 1 \tag{2}$$

However, using the width W_j of each R_j they can obtain the hiding capacity of two consecutive pixels by using:

$$t_i = \lfloor \log_2 W_j \rfloor \tag{3}$$

According to the above equation, t_i means the number of bits that can be hidden in each B_i. Read t_i bits from the binary secret bit-stream and it will be transform into its integer value b_i. For example, if t_i bits = "100", then the converted value of b_i = 4. Now, a new difference value d_i' can be calculated by adding l_j and b_i together where:

$$d_i' = \begin{cases} l_j + b_i, & \text{if } d_i \geq 0 \\ -(l_j + b_i), & \text{otherwise} \end{cases} \tag{4}$$

You should notice that the calculated d_i' will replace the original d_i. Finally, the secret data can be hidden into B_i by modifying its pixel values p_i and p_{i+1} as follows:

$$\begin{aligned} p_i' &= p_i - \lceil m/2 \rceil \\ p_{i+1}' &= p_{i+1} + \lfloor m/2 \rfloor \end{aligned} \tag{5}$$

where: $m = (d_i' - d_i)$. However, all the above steps are repeated for each sub-block until all secret data bits are hidden into the cover-image. Therefore, the stego-image is obtained.

Maleki, *et al* in [48] proposed an adaptive scheme based on Human Visual System (HVS), thus the pixels in edge regions can tolerate much more modifications than smooth regions. They use five secret keys, which are R_1, R_2, v_1, v_2 and T, where $v_1 \geq 1$, $v_2 \geq 1$ and ($v_1 + v_2 < 6$). First of all, two parameters K_r and K_c are generated using $H_r(R_1, v_1)$ and $H_c(R_2, v_2)$, respectively. Then, they calculate the average difference value D of 4-neighborhing pixels in a block as follows:

$$D = \frac{1}{3}\sum_{i=0}^{3}(y_i - y_{\min}), \text{where}: y_{\min} = \min\{y_0, y_1, y_2, y_3\} \tag{6}$$

According to a threshold secret key T, they classify the current block into edge region or smooth region. Pixels in edge regions are embedding with a larger number of bits than that belongs to smooth regions. On the other words, If $D \leq T$ that means the block belongs to smooth regions and then $Q = v_1$ bits will be embedded in that block. Otherwise, the block belongs to edge regions and then $Q = v_1 + v_2$. They have to determine if the current block is an error block or not by using this condition: $D \leq T, (y_{\max} - y_{\min}) > 2 \times T + 2$. If that condition is satis-

fy, then this block is called an error block and it is not used to embed secret data.

The main goal of the overlapping PVD (OPVD) method is to maximize the embedding capacity of the original PVD while maintaining acceptable image quality [49]. OPVD method embeds the secret data bits using singular pixels rather than embedding in pixel pairs using least significant bit (LSB) substitution. If the difference value of the pixel pair before and after embedding the secret data bits belongs to the same range, then the embedding procedure is implemented. Otherwise, there is no embedding for secret bits and the second pixel is adjusted. Despite the fact that this method conceals more secret data bits than PVD, its embedding capacity is limited since it still has a large number of unused pixels in embedding process. Moreover, using simple LSB approach and the adjustment process distorts the stego-image histogram. Therefore to increase the embedding capacity and improving the security of OPVD method while maintaining the image quality, [50] proposed a novel steganographic method based on OPVD. Like OPVD, the proposed method uses the difference of a two pixel block to recognize the smoothness and contrast in the current block. Then, it hides the secret bits in the second pixel based on the computed difference. After that, the second pixel is employed as first pixel in the next block and the embedding process is repeated. Moreover, a correction procedure is used to reduce the number of unused pixels.

To overcome the limitation of embedding capacity in PVD method, there are some PVD-based steganographic methods used a combination of PVD and LSB to dramatically enlarge the embedding capacity of secret data [13] [51]. Khodaei and Faez in [52] have proposed a steganographic method based on the PVD and modified LSB to embed the secret data within a greyscale cover-image. Their proposed method firstly divides all possible differences into lower and higher levels with a number of ranges. Secondly, it partitions the cover-image into non-overlapping blocks of three consecutive pixels and obtains the second pixel of each block as the base pixel. Next, by using LSB substitution followed by optimal pixel adjustment process (OPAP) which will described later, they embed k-bits of secret data in the base pixel. After that, they calculate the differences of pixel values between the base pixel and other two adjacent pixels in each block. Finally, they apply the modified PVD algorithm to embed secret data into the two pixels. In this method, each pixel embedded by modified PVD will conceal at least 3 bits, while in the original PVD each pixel pair will conceal at least 3 bits. Therefore, this method can embed large amount of secret data with maintaining acceptable visual quality of the stego-image.

Wang, *et al* in [53] proposed a high quality steganographic method based on PVD and modulus function, which is more secure against the RS detection attack and performs better than the PVD scheme. This scheme increased the peak signal-to-noise ratio (PSNR) values to 44.15 dB while concealed 51,219 bytes. It exploits the remainder of the two consecutive pixels to record the information of the embedded data, which achieves more flexibility, capable of deriving the optimal remainder of the two pixels at the least distortion. This method increased

the PSNR (up to 8.9%) more than the simple PVD method. To maintain the difference in the same range before and after embedding process, this method uses readjusting procedure to alter the reminder of the pixel pair.

Joo, *et al.* in [54] presented an enhancement on [53] method by embedding different amounts of secret data based on pixel-pair complexity. Tests in this method showed that the difference histogram had a shape closer to the cover-image which was hard to be detected by histogram analysis. Although this method improved the problems of the shapes in the difference histogram, its embedding capacity is not higher than Wang *et al.* method. In Joo *et al.* method, the embedding order is diverse for the odd and even embedding areas.

Chen in [55] introduced a PVD method using pixel pair matching (PPM). PPM [56] used two pixels as a unit for embedding a message digit S_B in B-ary notational system. In Chen method, the cover-image is partitioned into 2 × 2 embedding cells for embedding by random embedding arrangements. To increase the random embedding characteristic, two reference tables are created. This random mechanism raises the security of the embedded data from detection and other steganalysis attacks. The major contributions of this approach are that: (1) PPM was utilized thus more data was concealed than original PVD, (2) Effectively decreasing the falling-off-boundary problem by manipulating only on Pivot Embedding Unit (PEU). (3) The secret data was concealed based on two reference tables which raised the random characteristic and the visual quality. (4) This method is harder to be detected since its difference histogram demonstrates that the values of the stego-image are very close to the values of the cover-image. Comparison this method with [54], Chen scheme significantly had higher capacity and image quality.

3. The Proposed Algorithm

3.1. Embedding Phase

The embedding phase for our proposed algorithm used the same equations as employed in [41] [48] [52] [57]. The nitty gritty embedding steps for inserting secret data bits in a cover-image are described as follows:

Input: Cover-image *CI*, secret data *S*, and secret keys (R_1, R_2, α, β).

Output: Stego-image *SI*.

Step 1. The cover-image is dividing into non-overlapping two-pixel blocks $B = \{B_i | i = 1, 2, 3, \cdots, (W \times H)/2\}$. In each block, there are two neighboring pixels p_i and p_{i+1}, and their corresponding gray values are g_0 and g_1, respectively.

Step 2. Transform S into a binary bitstream S'.

Step 3. As explained previously, $H_r(R_1, \alpha)$ and $H_c(R_2, \beta)$ are used to generate two binary sets K_r and K_c, respectively.

Step 4. Calculate Q as follows:

$$Q = (\alpha + \beta) \tag{7}$$

Step 5. For the first pixel p_i in the block, $S_Q = Q$ bits of binary bitstream S'. After that, this S_Q is divided into two sub-sets S_{Q1} and S_{Q2}, where S_{Q1} has α bits

and S_{Q2} has β bits.

Step 6. Find the indices i and j using $S_{Q1} = K_{ri}$ and $S_{Q2} = K_{cj}$, where K_{ri} and K_{cj} are ith and jth binary elements in K_r and K_c sets, respectively.

Step 7. Compute y as follows:

$$y = 2^{\beta} \times (i-1) + j \tag{8}$$

Step 8. Generate a pixel group G which is a subset of the pixel intensity set $G = \{g_i \mid i = 1, 2, \cdots, n\}$ and is generated as follows:

$$f(p_i) = p_i \bmod n, \text{where } n = 2^Q \tag{9}$$

Thus, the pixel group G is an ordered set $\{p_i - f(p_i), p_i - f(p_i)+1, \cdots, p_i, p_i+1, p_i+n-f(p_i)-1\}$. Then, determine the corresponding stego-pixel from yth element of G where $p_i' = g_y$.

Step 9. For reducing perceptual distortion between the cover and stego images, we use "error reducing process". Let $L \in [-1, 0, 1]$ and $n = 2^Q$. After that, calculate p_i'' as in the following:

$$p_i'' = p_i' + L \times n \tag{10}$$

Consequently, we get three values for p_i''. We choose one p_i'' value, which is the closest value for the original value p_i. Thus, the final value for the corresponding stego-pixel p_i' after error reducing process will be the chosen p_i''.

Step 10. Until now, we embed Q bits in the first pixel in the block B_i. Now, compute the difference value between p_i' and the second pixel p_{i+1} in B_i as follows:

$$d_i = |p_{i+1} - p_i'| \tag{11}$$

Step 11. Find the corresponding sub-range $R_i \in [l_i, u_i]$ for the resulted difference value d_i from the dividing range table that is shown in Figure 1. Where l_i and u_i are the lower and higher bounds for sub-range R_i.

Step 12. If R_i belongs to the lower-level, then 3 secret data bits will be embedded in the second pixel p_{i+1}. Unless 4 secret data bits will be embedded in the second pixel p_{i+1}. By considering t_i is the number of embedded secret data bits, then, read t_i bits from the binary bitstream S' and convert it into its decimal value s_t.

Step 13. Compute the new difference value dd_i as follows:

$$dd_i = l_i + s_t. \tag{12}$$

Step 14. Calculate the stego pixel value p_{i+1}' for the second pixel p_{i+1} in the block B_i using the following formula:

$$p_{i+1}' = \begin{cases} p_i' - dd_i, & \text{if } |p_{i+1} - p_{i+1}'| < |p_{i+1} - (p_i' + dd_i)| \text{ and } 0 \le p_{i+1}' \le 255 \\ p_i' + dd_i, & \text{otherwise} \end{cases} \tag{13}$$

Lower-level		Higher-Level			
$R_1 = [0, 7]$	$R_2 = [8, 15]$	$R_3 = [16, 31]$	$R_4 = [32, 63]$	$R_5 = [64, 127]$	$R_6 = [128, 255]$
$t_i = 3$ bits		$t_i = 4$ bits			

Figure 1. Dividing range table R.

Step 15. Now, we are getting the stego-block consisting p_i' and p_{i+1}'. After that, repeat the steps from 5 to 15 for the next block until all secret data bits are completely embedded and the stego-image SI is obtained.

Figure 2 shows the flowchart of the embedding process in MF-PVD algorithm.

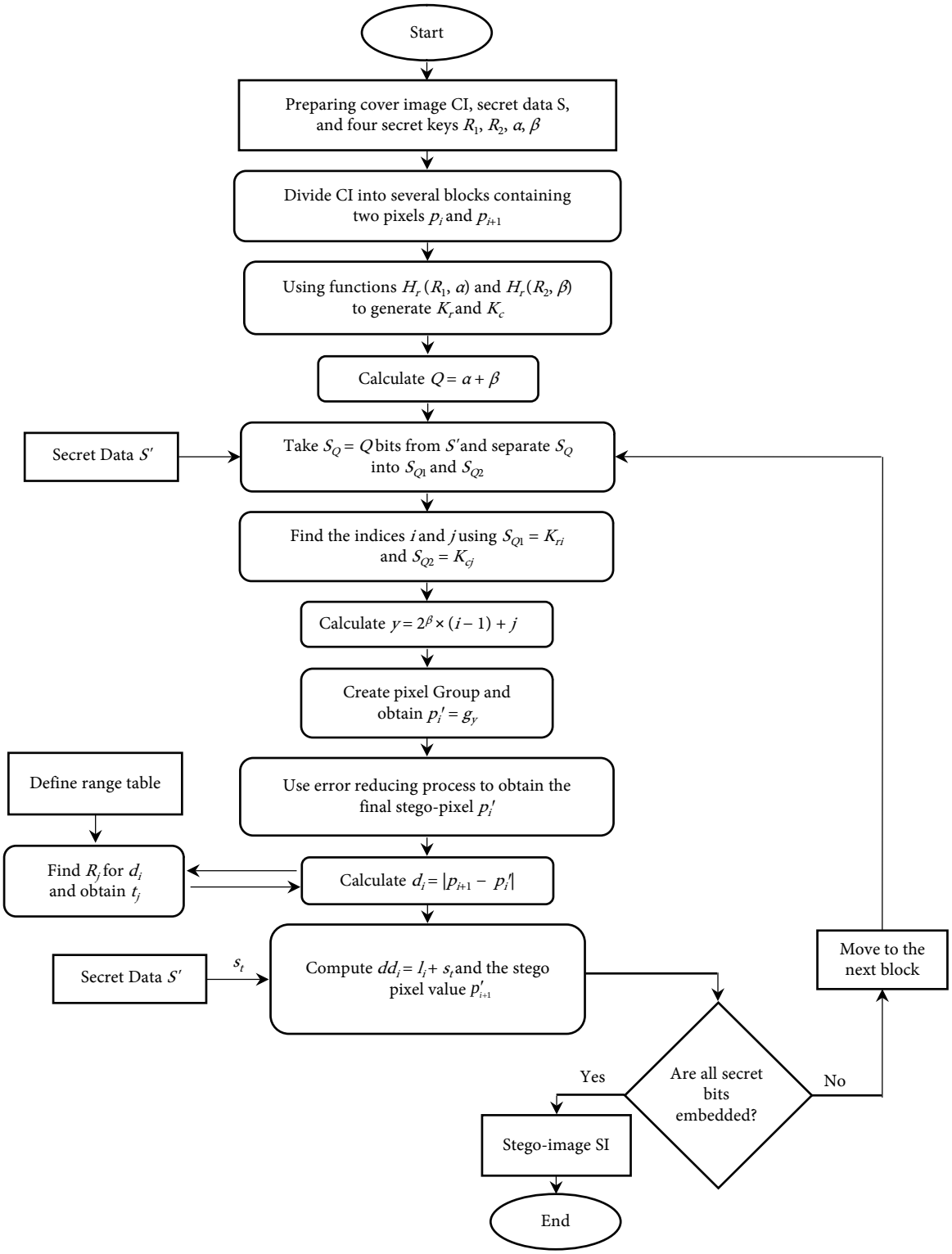


Figure 2. Embedding process in MF-PVD algorithm.

Respect to the side information used in MF-PVD algorithm, they are offline, except the amount of the payload which takes 32 bits from the embedding capacity.

- Embedding Phase: *Example*

An illustration of the data embedding phase is shown in **Figures 3(a)-(h)**. Suppose that the simple block is comprised by p_1 and p_2 and their corresponding grey values are (47, 48) as shown in **Figure 3(a)**, and the bitstream of secret data is "1000011", as shown in **Figure 3(b)**. Also, assume $\alpha = 1$, $\beta = 3$, $R_1 = 2$ and $R_2 = 2020$.

- Before the actual embedding, we generate the K_r using H_r (2, 1) and the K_c using H_c (2020, 3). All permutations generated by H_r and H_c for K_r and K_c, respectively, are shown in **Figure 3(c)**, **Figure 3(d)**. Thus, $K_r = \{1, 0\}$ and $K_c = \{001, 010, 101, 000, 011, 100, 110, 111\}$.
- The first pixel is embedded with $Q = 4$ bits of secret data. Thus, $S_Q =$ "1000" will be separated into two sub-strings $S_{Q1} =$ "1" and $S_{Q2} =$ "000".
- After that, we find the indices i and j where $S_{Q1} = K_{ri}$ and $S_{Q2} = K_{cj}$. Therefore, $i = 1$ and $j = 4$.
- Using the values of i, j and β to compute $y = 2^{\beta} \times (i - 1) + j = 2^3 \times (1 - 1) + 4 = 4$.

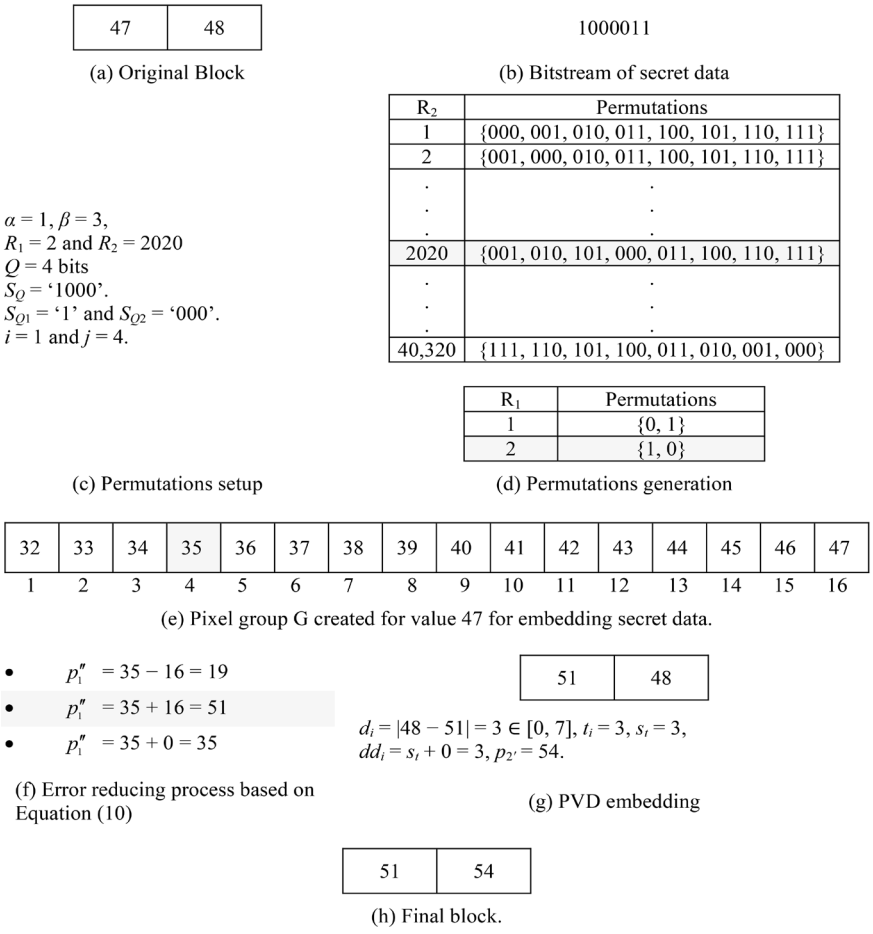


Figure 3. (a)-(h): An example of the embedding process in MF-PVD algorithm.

- Next, the pixel group G is created as shown in Figure 3(e). Therefore, the stego-pixel p_1' can be obtained from the *yth* element of G (*i.e.* $p_1' = g_4 = 35$).
- To reduce the distortion, we use the error reducing process. As shown in Figure 3(f), we will get three values for p_1', thus, we choose the closest one to the original pixel value 47, which is 51 (*i.e.* $p_1' = 51$).
- Now, we compute the difference value between p_1' and p_2 as follows: $d = |48 - 51| = 3 \in R_1$ where $R_1 = [0, 7]$ and $t_1 = 3$ bits. Thus, $s_t = 3$.
- Next, the new difference value is computed as follows: $dd = s_t + 0 = 3$
- Finally, according to step 14, the stego-pixel $p_2' = p_1' + dd = 51 + 3 = 54$

Hence, the block after embedding the secret data will be as shown in Figure 3(h).

3.2. Extracting Phase

The details of extracting phase to extract the secret data S are described as follows:

Input: Stego-image SI, and secret keys (R_1, R_2, α, β).

Output: Secret data S.

Step 1. Similar to the embedding phase, first of all, partition the SI into two-pixel non-overlapping blocks. In each block, there are two neighboring pixels (p_i', p_{i+1}').

Step 2. Using $H_r(R_1, \alpha)$ and $H_c(R_2, \beta)$ to generate two binary sets K_r and K_c, respectively.

Step 3. Through sets K_r and K_c, form a Cartesian product $K_r \oplus K_c$. $K_r \oplus K_c$ creates an ordered set of combinations of K_r and K_c with $2^\alpha \times 2^\beta = 2^{\alpha+\beta}$ elements. Each component of the variant Cartesian product $K_r \oplus K_c$ is binary string concatenation which includes the two binary strings K_{ri} and K_{cj} jointly to form one string which has the length $(\alpha + \beta)$ bits

$$K_r \oplus K_c = \left\{ K_{ri} \| K_{cj} \middle| K_{ri} \in K_r \text{ and } K_{cj} \in K_c, i = 1, 2, \cdots, 2^\alpha, j = 1, 2, \cdots, 2^\beta \right\} \quad (14)$$

For example, assume $\alpha = 1$, $\beta = 3$, $R_1 = 2$ and $R_2 = 2020$. We generate the K_r using $H_r(2, 1)$ and the K_c using $H_c(2020, 3)$. Thus, $K_r = \{1, 0\}$ and $K_c = \{001, 010, 101, 000, 011, 100, 110, 111\}$. We produce the variant Cartesian product $K_r \oplus K_c$ as follows: {1001, 1010, 1101, 1000, ⋯, 0011, 0100, 0110, 0111}

Step 4. Calculate Q as was in the Equation (7)

Step 5. For the first pixel p_i' in the block create the pixel group G and find the position y of p_i', since the stego pixel p_i' equals g_y (*i.e.* $p_i' = g_y$, where $n = 2^Q$).

Step 6. Use the Cartesian product of K_r and K_c($K_r \oplus K_c$) to extract the *yth* element which is the secret embedded bits with Q bits, we called this first piece of secret data by S_Q.

Step 7. Find the difference value between the first and second pixels in the block where

$$d_i' = |p_{i+1}' - p_i'| \quad (15)$$

Step 8. Find the corresponding range $R_i \in [l_i, u_i]$ for the resulted difference value d_i' and compute t_i.

Step 9. Extract the second piece of secret data S_t by using:

$$S_t = d_i' - l_i \tag{16}$$

Step 10. Transform S_t into its binary value.

Step 11. Move to next block and repeat Steps from 5 to 11 until all the pieces of secret data are completely extracted. Then, concatenate all the pieces of secret data sub-bitstreams in order to recover the required hidden secret data *S*.

The error reducing process in our algorithm can work correctly since it is do not change the hidden secret data. To prove that, remember step 5 in the extracting phase. We mentioned that $p_i' = g_y$ where $n = 2^Q$, if we applying the following equation: $p_i'' = p_i' + L \times n$, where $L \in [-1, 0, 1]$, we will get three values ($p_i' - 2^Q$), (p_i') and ($p_i' + 2^Q$).

All these resulted values have the same reminder to *n*. Consequently, in the extracting phase, using each of these three values will lead to extract the same secret data correctly. Detection process for secret data in our method is very difficult to any unauthorized users due to existing many permutations (K_r has $2^\alpha!$), (K_c has $2^\beta!$) and ($K_r \oplus K_c$ has $2^\alpha! \times 2^\beta!$). Thus, an attacker will face more difficult in guessing the secret data. **Figure 4** shows the flowchart of the extracting process in MF-PVD algorithm.

4. Experimental Results and Discussions

4.1. Simulation Setup: Simulation Parameters and Performance Metrics

All our experiments are developed using MATLAB 8.2.0.701 (R2013b) software on Windows 7 platform with an Intel Core i7-4600U CPU working at 2.1 GHz with a 4 MB cache and 4 GB RAM. We used different benchmark gray level images with size 512 × 512 from various databases such as (*USC-SIPI Image Database*) and (*UWATERLOO-LINKS Image Repository*) [58] [59]. Also, we used various formats for images, such as: *Tiff*, *Jpg*, *Bmp*, and *Gif*. In addition, we use *randseq* ( ) function to generate random secret message to be embedded in the cover image.

The effectiveness of our proposed algorithm is verified using different performance metrics such as PSNR and structural similarity index measure (SSIM). In general, PSNR and SSIM are used evaluate the overall image quality. PSNR is computed using the following equation [41] [60]-[68]:

$$PSNR = 10 \times \log_{10}\left(\frac{(I_{\max})^2}{MSE}\right) dB \tag{17}$$

where $I_{\max}$ equals to 255 for 8-bit gray level images, which means the maximum intensity value of each pixel. Mean square error (MSE) is calculated using:

$$MSE = \frac{1}{MN} \times \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(x_{ij} - x_{ij}'\right)^2 \tag{18}$$

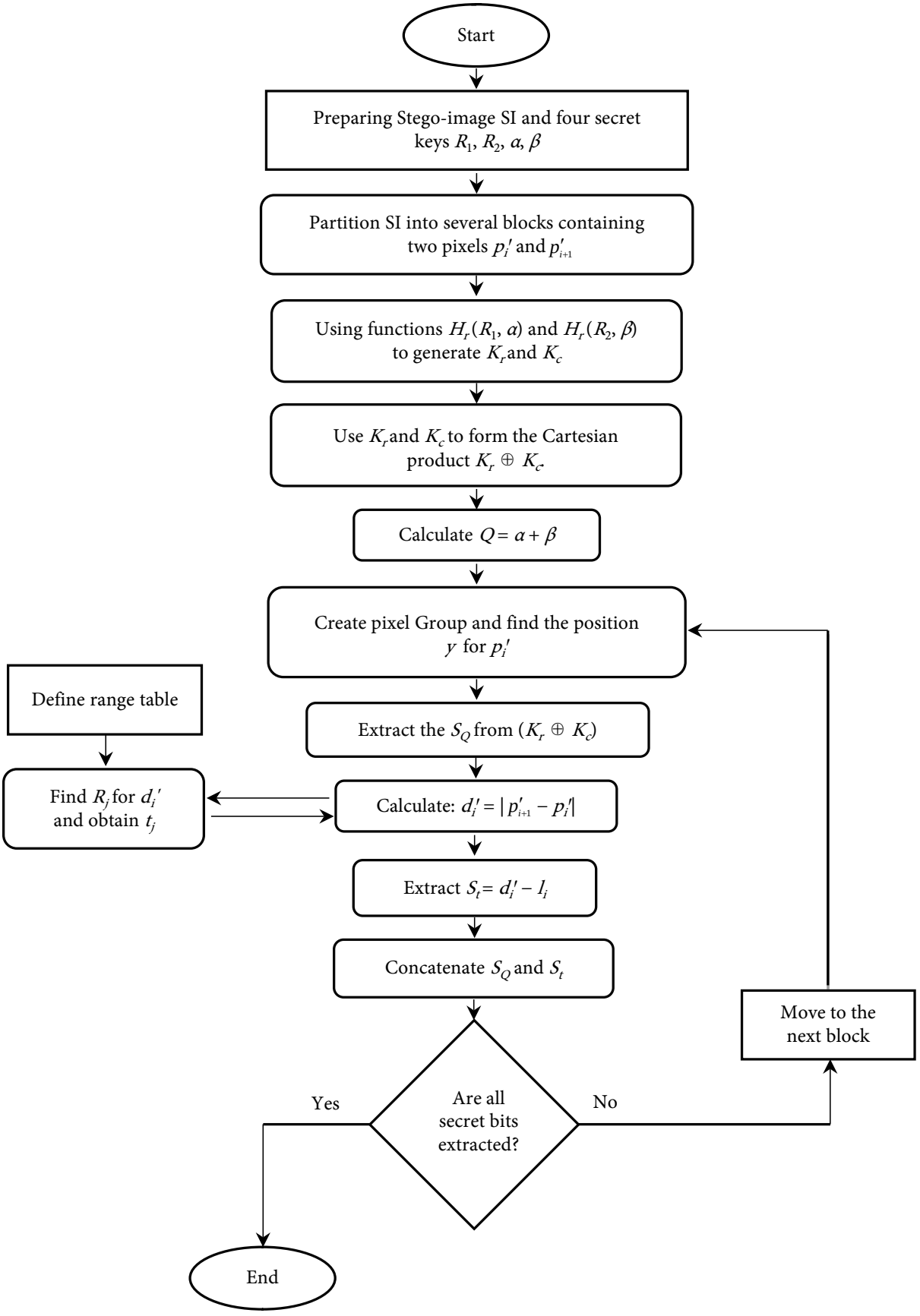


Figure 4. Extracting process in MF-PVD algorithm.

where MN is the total number of pixels for both cover and stego images. x_{ij} and x'_{ij} represent the pixels in the cover image and stego image, respectively. SSIM is calculated as follows [69] [70]:

$$SSIM(x,y)=\frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \tag{19}$$

where $C_1=(k_1L)^2$ and $C_2=(k_2L)^2$, which are two constants to stabilize the division when the mean and variance get close to zero. L represents the maximum possible value for image pixel, $L=2^{Nbpp}-1$ (*where* N_{bpp} *is the number of bits per pixel*). μ_x and σ_x^2 denote the mean and variance of x, respectively. μ_y and σ_y^2 denote the mean and variance of y, respectively, σ_{xy} refers to the covariance of x and y. The value of SSIM lies in the interval [*zero*, *one*]. The value "*one*" means that both images, cover image and stego image, are precisely the same, and the value "*zero*" means that they are absolutely unrelated. However, for each image, there are several SSIM indexes where each one is calculated within (11 × 11) local window using a certain circular-symmetric Gaussian weighting value (between *zero* and *one*) and the final SSIM image index is the average of these indexes.

4.2. Experimental Results and Discussions

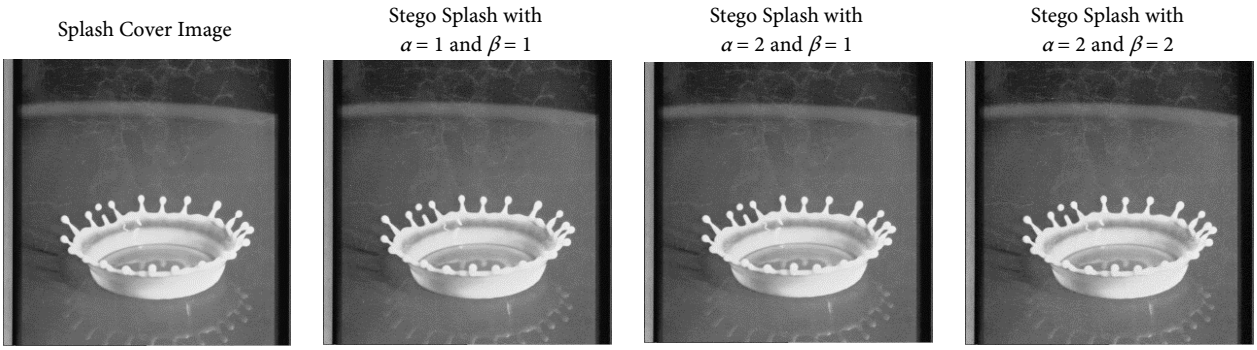
The maximum embedding capacity (EC) in our algorithm can be computed by using the following equation:

$$EC=(N_{fp}\times(\alpha+\beta))+(N_{sp1}\times 3)+(N_{sp2}\times 4) \tag{20}$$

where N_{fp} is the number of first pixel blocks, N_{sp1} is the number of the second pixel blocks that belongs to the lower-level and N_{sp2} is the number of the second pixel blocks that belongs to the higher-level.

We have implemented using a series of α and β secret keys. The value we got from adding α and β values will be embedded in the first pixel in each block (*i.e.* α + β bits). Figure 5 and Figure 6 demonstrate the experimental results of six stego-images resulted by our algorithm with different values for α and β on Elaine and Splash images.

Figure 5. Results of MF-PVD on Stego-Elaine with different secret keys.



EC (Bits)	658,841	790,005	921,888
SSIM	0.9914	0.9861	0.9685
PSNR	38.16	37.29	34.96

Figure 6. Results of MF-PVD on Stego-Splash with different secret keys.

Tables 1(a)-1(c) shows the results of our adaptive algorithm in terms of the maximum embedding capacity, SSIM and PSNR values. As shown in Table 1, our proposed algorithm is scalable and flexible, so that larger values for α and β improve the embedding capacity whereas a lower values of α and β improve the stego-image quality. Moreover, it is noteworthy that the complexity time presented by our algorithm almost is $O(n^2)$, which is less than or equals to 59 s.

Our algorithm outperforms Maleki, *et al.* scheme [48] since we have not error blocks which is not used to embed secret data. On the other hand, using the PVD method for embedding the secret data in the second pixel according to the specific range table significantly increases the capability of embedding. Therefore, our method is extremely superior Maleki *et al.* scheme in terms of the embedding capacity. Table 2(a), Table 2(b) demonstrates a comparison between our algorithm and Maleki *et al.* algorithm. As shown in Table 2, in all different values of α and β, our method provides the highest embedding capacities. In addition to improve the embedding capacity, our method does not degrade much on the visual quality of the stego-image. The embedding rate and the visual quality of the stego-image in our algorithm can be modified depending on the requirements of the practical applications. In other words, if we need high embedding rate, we have to choose larger values for α and β, but if we need high visual quality, we have to choose smaller values for α and β. Compared Maleki *et al.* algorithm with our proposed algorithm in terms of the EC, the average improvement ratio (AIR) is about 47%, 20%, 25% and 12% when using ($\alpha = 1$ and $\beta = 1$), ($\alpha = 2$ and $\beta = 1$), ($\alpha = 2$ and $\beta = 2$) and ($\alpha = 3$ and $\beta = 1$), respectively. Unfortunately, this will be at the cost of decreasing PSNR values. The average degradation ratio (ADR) is about 18%, 10%, 5% and 8%, respectively. Although, there are decreasing in PSNR values, the stego image visual quality does not show any distortion to be suspected by unauthorized observers.

Results shown in Table 3(a), Table 3(b) demonstrates that our algorithm outperforms to [50] and [52] algorithms in terms of the embedding capacity. The AIR is about 12% and 2%, respectively. Moreover, our algorithm provides better PSNR values in most stego images than El-Alfy and Al-Sadi. Although our

Table 1. (a)-(c): Experimental results of MF-PVD algorithm.

Cover image name	$\alpha = 1, \beta = 1$				$\alpha = 1, \beta = 2$			
	EC (bits)	ER (bpp)	SSIM	PSNR (dB)	EC (bits)	ER (bpp)	SSIM	PSNR (dB)
				Panel a				
Lena	667,947	2.548	0.9943	39.26	799,802	3.051	0.9912	38.11
Peppers	665,818	2.530	0.9944	38.58	797,937	3.044	0.9919	37.57
Boat	674,691	2.574	0.9938	37.32	806,296	3.076	0.9918	36.63
Jet	665,409	2.538	0.9929	38.28	796,637	3.039	0.9883	37.48
Splash	658,841	2.513	0.9914	38.16	790,046	3.014	0.9861	37.32
Baboon	699,727	2.669	0.9956	35.97	831,709	3.173	0.9946	35.43
Tiffany	665,562	2.539	0.9924	39.29	796,903	3.040	0.9924	38.25
Elaine	669,477	2.554	0.9947	39.68	801,186	3.056	0.9927	38.54
Couple	675,311	2.576	0.9952	37.97	806,691	3.077	0.9934	37.14
Average	671,420	2.560	0.9939	38.28	803,023	3.063	0.9914	37.39
Cover image name	$\alpha = 1, \beta = 3$				$\alpha = 2, \beta = 1$			
	EC (bits)	ER (bpp)	SSIM	PSNR(dB)	EC (bits)	ER (bpp)	SSIM	PSNR (dB)
				Panel b				
Lena	935,756	3.570	0.9808	35.35	799,796	3.051	0.9912	38.15
Peppers	934,567	3.565	0.9825	35.12	797,927	3.044	0.9917	37.59
Boat	940,248	3.587	0.9847	34.55	806,216	3.075	0.9919	36.61
Jet	928,517	3.542	0.9706	35.05	796,643	3.039	0.9882	37.45
Splash	921,817	3.516	0.9683	34.95	790,005	3.014	0.9861	37.29
Baboon	960,037	3.662	0.9907	33.79	831,288	3.171	0.9946	35.46
Tiffany	929,241	3.545	0.9746	35.54	796,893	3.040	0.9883	38.25
Elaine	935,520	3.569	0.9854	35.69	801,306	3.057	0.9926	38.51
Couple	939,471	3.584	0.9873	34.94	806,727	3.077	0.9935	37.10
Average	936,130	3.571	0.9805	35.00	802,978	3.063	0.9909	37.38
Cover image name	$\alpha = 2, \beta = 2$				$\alpha = 3, \beta = 1$			
	EC (bits)	ER (bpp)	SSIM	PSNR (dB)	EC (bits)	ER (bpp)	SSIM	PSNR (dB)
				Panel c				
Lena	935,830	3.570	0.9808	35.39	935,721	3.569	0.9807	35.38
Peppers	930,307	3.549	0.9826	35.24	930,439	3.549	0.9825	35.23
Boat	940,185	3.587	0.9843	34.55	940,190	3.587	0.9846	34.59
Jet	928,554	3.542	0.9701	35.02	928,518	3.542	0.9705	35.02
Splash	921,888	3.517	0.9685	34.96	921,880	3.517	0.9686	34.96
Baboon	960,035	3.662	0.9908	33.84	964,208	3.678	0.9908	33.78
Tiffany	929,242	3.545	0.9746	35.53	929,185	3.545	0.9746	35.33
Elaine	935,455	3.568	0.9854	35.68	935,387	3.568	0.9853	35.69
Couple	939,466	3.584	0.9873	34.87	939,535	3.584	0.9873	34.88
Average	935,662	3.569	0.9805	35.01	936,118	3.571	0.9805	34.98

Table 2. (a) (b): Comparison of the results between Maleki, *et al.* scheme [48] and MF-PVD algorithm.

Cover image name	Maleki 1 - 2, T = 3		Our method $\alpha = 1, \beta = 1$		Maleki 2 - 3, T = 5		Our method $\alpha = 2, \beta = 1$	
	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)
				Panel a				
Lena	455,908	46.66	667,947	39.26	661,852	41.89	799,796	38.15
Peppers	467,264	46.53	665,818	38.58	670,300	41.56	797,927	37.59
Boat	486,960	46.50	674,691	37.32	703,840	41.18	806,216	36.61
Jet	393,888	47.54	665,409	38.28	613,560	43.05	796,643	37.45
Splash	404,948	47.13	658,841	38.16	598,144	43.06	790,005	37.29
Baboon	514,252	46.36	699,727	35.97	758,984	40.81	831,288	35.46
Tiffany	436,584	46.87	665,562	39.29	642,692	42.24	796,893	38.25
Elaine	493,948	46.51	669,477	39.68	718,440	41.06	801,306	38.51
Couple	479,912	46.53	675,311	37.97	693,864	41.43	806,727	37.10
Average	459,296	46.74	671,420	38.28	673,520	41.81	802,978	37.38
Cover image name	Maleki 2 - 4, T = 6		Our method $\alpha = 2, \beta = 2$		Maleki 3 - 4, T = 15		Our method $\alpha = 3, \beta = 1$	
	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)
				Panel b				
Lena	760,952	36.41	935,830	35.39	834,264	38.37	935,721	35.38
Peppers	766,792	36.06	930,307	35.24	824,100	38.73	930,439	35.23
Boat	800,040	35.48	940,185	34.55	846,968	37.82	940,190	34.59
Jet	680,088	38.23	928,554	35.02	824,148	38.88	928,518	35.02
Splash	633,016	38.80	921,888	34.96	802,052	39.71	921,880	34.96
Baboon	800,032	35.52	960,035	33.84	926,840	35.98	964,208	33.78
Tiffany	726,376	37.12	929,242	35.53	823,408	38.78	929,185	35.33
Elaine	800,032	35.23	935,455	35.68	831,240	38.03	935,387	35.69
Couple	800,032	35.88	939,466	34.87	856,396	37.68	939,535	34.88
Average	751,929	36.53	935,662	35.01	841,046	38.22	936,118	34.98

Table 3. (a) (b): Comparison of the results between our MF-PVD algorithm against [50] and [52] methods.

Cover image name	El-Alfy and Al-Sadi method [50]		Our method $\alpha = 3, \beta = 1$	
	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)
	Panel a			
Lena	820,150	35.56	935,721	35.38
Peppers	815,337	35.05	930,439	35.23
Boat	845,209	33.34	940,190	34.59
Baboon	921,399	29.90	964,208	33.78
Elaine	820,296	36.42	935,387	35.69
Average	844,478	34.05	941,189	34.93
Cover image name	Khodaei and Faez method using first range table and k = 3 bits [52]		Our method $\alpha = 2, \beta = 1$	
	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)
	Panel b			
Lena	788,407	37.35	799,796	38.15
Peppers	787,506	35.68	797,927	37.59
Boat	792,319	36.09	806,216	36.61
Jet	788,685	36.81	796,643	37.45
Baboon	806,411	34.45	831,288	35.46
Tiffany	787,389	37.63	796,893	38.25
Average	791,786	36.34	804,794	37.25

algorithm does not greatly increase the embedding capacity than Khodaei and Faez, our algorithm does not degrade on the image quality and it provides higher level of security. The level of security in our proposed algorithm is high, due to two reasons, which are: 1) We have two methods to individually embed each pixel in a block, 2) Our algorithm provides hard detection for the hidden secret data bits due to existing: many permutations and dividing range table.

Table 4(a), **Table 4(b)** shows the comparison of the results between our algorithm against [53] [54] [55] algorithms. In fact, our algorithm is superior to these three approaches in two features, namely embedding capacity and level of security. The AIR in terms of the embedding capacity is about 62%, 61% and 55%, respectively. However, as inevitable result of the increasing in the embedding capacity, the PSNR is decreased. The ADR in terms of the PSNR is about 11%, 11% and 19%, respectively. Our algorithm has higher level of security than these three methods for the same reasons listed previously.

5. Conclusion and Future Work

We have proposed a new block-based steganographic algirthm using PVD and modulus function techniques, namely, MF-PVD. To evaluate the performance of MF-PVD algorithm, we compare it with six pertinent state-of-art algorithms,

Table 4. (a) (b): Comparison of the results between MF-PVD algorithm, [53], [54] and [55] methods.

Cover image name	Wang, *et al.* method [53]		Our method $\alpha = 1$, $\beta = 1$	
	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)
	Panel a			
Lena	409,752	44.15	667,947	39.26
Peppers	407,256	43.28	665,818	38.58
Boat	421,080	42.14	674,691	37.32
Jet	421,080	42.14	665,409	38.28
Splash	389,459	44.34	658,841	38.16
Baboon	457,168	40.32	699,727	35.97
Tiffany	407,360	43.80	665,562	39.29
Elaine	408,592	44.74	669,477	39.68
Couple	412,824	43.25	675,311	37.97
Average	414,952	43.13	671,420	38.28
Cover image name	Joo, *et al.* method [54]		Chen method [55]	
	EC (bits)	PSNR (dB)	EC (bits)	PSNR (dB)
	Panel b			
Lena	409,785	43.90	419,340	47.50
Peppers	407,256	43.10	436,808	47.30
Boat	421,083	41.90	429,744	47.40
Jet	409,818	43.30	426,320	47.40
Baboon	457,169	40.20	459,792	47.10
Elaine	408,594	44.80	434,128	47.40
Average	418,951	42.87	434,355	47.35

which are existed in [48] [50] [52] [53] [54] [55]. As a matter of fact, our MF-PVD algorithm is outstanding to these mentioned methods in two main features, the embedding capacity and the security. In fact, the security of our algorithm is high due to generating many permutations and existing the dividing range table.

Many trends can be given for further improvements to the proposed algorithm. The algorithm's framework can be extended to the RGB color images for improving the capability of embedding. Moreover, it can be a good addition to develop an approach that takes into account the hybrid domain. As well, there are future plans to develop modulus function-based schemes for another media such as audios and videos.

References

[1] Darabkh, K.A., Albtoush, W.Y. and Jafar, I.F. (2016) Improved Clustering Algorithms for Target Tracking in Wireless Sensor Networks. *Journal of Supercomput-*

ing, Online.

[2] Darabkh, K.A. and Aygün, R.S. (2007) TCP Traffic Control Evaluation and Reduction over Wireless Networks Using Parallel Sequential Decoding Mechanism. *EURASIP Journal on Wireless Communications and Networking*, **2007**, Article ID: 52492. https://doi.org/10.1155/2007/52492

[3] Darabkh, K.A. (2010) Queuing Analysis and Simulation of Wireless Access and End Point Systems Using Fano Decoding. *Journal of Communications*, **5**, 551-561. https://doi.org/10.4304/jcm.5.7.551-561

[4] Darabkh, K.A., Khalifeh, A.F., Jafar, I.F., Bathech, B.A. and Sabah, S.W. (2013) Efficient DTW-Based Speech Recognition System for Isolated Words of Arabic Language. *Proceedings of International Conference on Electrical and Computer Systems Engineering*, 77, 689-692.

[5] Darabkh, K.A., Khalifeh, A.F., Jafar, I.F., Bathech, B.A. and Sabah, S.W. (2013) A Yet Efficient Communication System with Hearing-Impaired People Based on Isolated Words of Arabic Language. *IAENG International Journal of Computer Science*, **40**, 183-193.

[6] Darabkh, K.A. (2011) Evaluation of Channel Adaptive access Point System with Fano Decoding. *International Journal of Computer Mathematics*, **88**, 916-937. https://doi.org/10.1080/00207160.2010.485249

[7] Darabkh, K.A. (2015) Fast and Upper Bounded Fano Decoding Algorithm: Queuing Analysis. *Transactions on Emerging Telecommunications Technologies*, Online.

[8] Hawa, M., Darabkh, K.A., Al-Zubi, R. and Al-Sukkar, G. (2016) A Self-Learning MAC Protocol for Energy Harvesting and Spectrum Access in Cognitive Radio Sensor Networks. *Journal of Sensors*, **2016**, Article ID: 9604526. https://doi.org/10.1155/2016/9604526

[9] Darabkh, K.A., Abu-Jaradeh, B.N. and Jafar, I.F. (2011) Incorporating Automatic Repeat Request and Thresholds with Variable Complexity Decoding Algorithms over Wireless Networks: Queuing Analysis. *IET Communications Journal*, **5**, 1377-1393. https://doi.org/10.1049/iet-com.2010.0698

[10] Darabkh, K.A., Jafar, I., Al Sukkar, G., Abandah, G. and Al-Zubi, R. (2012) An Improved Queuing Model for Packet Retransmission Policy and Variable Latency Decoders. *IET Communications Journal*, **6**, 3315-3328. https://doi.org/10.1049/iet-com.2012.0410

[11] Hawa, M., Darabkh, K.A., Khalaf, L.D. and Rahhal, J.S. (2015) Dynamic Resource Allocation Using Load Estimation in Distributed Cognitive Radio Systems. *AEÜ—International Journal of Electronics and Communications*, **69**, 1833-1846. https://doi.org/10.1016/j.aeue.2015.09.008

[12] Ismail, S.S., Al Khader, A.I. and Darabkh, K.A. (2015) Static Clustering for Target Tracking in Wireless Sensor Networks. *Global Journal on Technology*, **8**, 167-173.

[13] Darabkh, K.A., Jafar, I.F., Al-Zubi, R.T. and Hawa, M. (2014) An Improved Image Least Significant Bit Replacement Method. *Proceedings of the* 37*th IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics*, Opatija, 26-30 May 2014, 1182-1186. https://doi.org/10.1109/mipro.2014.6859747

[14] Al-Zubi, R., Krunz, M., Al-Sukkar, G., Hawa, M. and Darabkh, K.A. (2014) Packet Recycling and Delayed ACK for Improving the Performance of TCP over MANETs. *Wireless Personal Communications*, **75**, 943-963. https://doi.org/10.1007/s11277-013-1401-8

[15] Darabkh, K.A. and Alsukour, O. (2015) Novel Protocols for Improving the Perfor-

mance of ODMRP and EODMRP over Mobile Ad Hoc Networks. *International Journal of Distributed Sensor Networks*, **2015**, Article ID: 348967. https://doi.org/10.1155/2015/348967

[16] Darabkh, K.A., Ibeid, H., Jafar, I.F., Al-Zubi, R.T. (2016) A Generic Buffer Occupancy Expression for Stop-and-Wait Hybrid Automatic Repeat Request Protocol over Unstable Channels. *Telecommunication Systems*, **63**, 205-221. https://doi.org/10.1007/s11235-015-0115-5

[17] Darabkh, K.A. and Pan, W.D. (2006) Stationary Queue-Size Distribution for Variable Complexity Sequential Decoders with Large Timeout. *Proceedings of the* 44*th ACM Southeast Conference*, Melbourne, 10-12 March 2006, 331-336.

[18] Al-Mistarihi, M.F., Mohaisen, R., Sharaqa, A., Shurman, M.M. and Darabkh, K.A. (2015) Performance Evaluation of Multiuser Diversity in Multiuser Two-Hop Cooperative Multi-Relay Wireless Networks using MRC over Rayleigh Fading Channels. *International Journal of Communication Systems*, **28**, 71-90. https://doi.org/10.1002/dac.2640

[19] Shurman, M., Al-Shua'b, B., Alsaedeen, M., Al-Mistarihi, M.F. and Darabkh, K. (2014) N-BEB: New Backoff Algorithm for IEEE 802.11 MAC Protocol. *Proceedings of* 37*th IEEE International Convention on Information and Communication Technology*, *Electronics and Microelectronics* (*MIPRO* 2014), Opatija, Croatia, May 2014, 540-544.

[20] Al-Zubi, R., Hawa, M., Al-Sukkar, G. and Darabkh, K.A. (2014) Markov-Based Distributed Approach For Mitigating Self-Coexistence Problem in IEEE 802.22 WRANs. *The Computer Journal*, **57**, 1765-1775. https://doi.org/10.1093/comjnl/bxt092

[21] Shurman, M., Awad, N., Al-Mistarihi, M.F. and Darabkh, K.A. (2014) LEACH Enhancements for Wireless Sensor Networks Based on Energy Model. *Proceedings of the* 2014 *IEEE International Multi-Conference on Systems*, *Signals & Devices*, *Conference on Communication & Signal Processing*, Castelldefels, 11-14 February 2014, 1-4.

[22] Shurman, M., Al-Mistarihi, M., Mohammad, A., Darabkh, K. and Ababnah, A. (2013) Hierarchical Clustering Using Genetic Algorithm in Wireless Sensor Networks. *Proceedings of* 36*th IEEE International Convention on Information and Communication Technology*, *Electronics and Microelectronics*, Opatija, 20-24 May 2013, 479-483.

[23] Darabkh, K.A., Al-Dhamari, A.K. and Jafar, I.F. A New Steganographic Algorithm Based on Multi Directional PVD and Modified LSB. To Appear in Information Technology and Control, Kaunas University of Technology, Kaunas.

[24] Jafar, I., Darabkh, K.A. and Saifan, R. (2016) SARDH: A Novel Sharpening-Aware Reversible Data Hiding Algorithm. *Journal of Visual Communication and Image Representation*, **39**, 239-252. https://doi.org/10.1016/j.jvcir.2016.06.002

[25] Jafar, I., Darabkh, K.A., Saifan, R. and Al-Zubi, R. (2016) An Efficient Reversible Data Hiding Algorithm Using Two Steganographic Images. *Signal Processing*, **128**, 98-109. https://doi.org/10.1016/j.sigpro.2016.03.023

[26] Jafar, I.F., Darabkh, K.A., Al-Zubi, R.T. and Nam'neh, R. (2016) Efficient Reversible Data Hiding Using Multiple Predictors. *The Computer Journal*, **59**, 423-438. https://doi.org/10.1093/comjnl/bxv067

[27] Jafar, I., Hiary, S. and Darabkh, K.A. (2014) An Improved Reversible Data Hiding Algorithm Based on Modification of Prediction Errors. *Proceedings of* 6*th International Conference on Digital Image Processing*, Athens, 5-6 April 2014, 91591U-91591U-6.

[28] Darabkh, K.A., Jafar, I.F., Al-Zubi, R.T. and Hawa, M. (2015) A New Image Steganographic Approach for Secure Communication Based on LSB Replacement Method. *Information Technology and Control*, **44**, 315-328. https://doi.org/10.5755/j01.itc.44.3.8949

[29] Darabkh, K.A. (2014) Imperceptible and Robust DWT-SVD-Based Digital Audio Watermarking Algorithm. *Journal of Software Engineering and Applications*, **7**, 859-871. https://doi.org/10.4236/jsea.2014.710077

[30] Cox, I., Miller, M., Fridrich, J. and Kalker, T. (2007) Digital Watermarking and Steganography. Morgan Kaufmann Publishers Inc., San Francisco.

[31] Cheddad, A., Condell, J., Curran, K. and Kevitt, P. (2010) Digital Image Steganography: Survey and Analysis of Current Methods. *Signal Processing*: *Image Communication*, **90**, 727-752. https://doi.org/10.1016/j.sigpro.2009.08.010

[32] Saini, S. and Brindha, K. (2014) Improved Data Embedding into Images Using Histogram Shifting. *International Journal of Emerging Research in Management & Technology*, **3**, 83-86.

[33] Kefa, R. (2004) Steganography—The Art of Hiding Data. *Information Technology Journal*, **3**, 245-269. https://doi.org/10.3923/itj.2004.245.269

[34] Lin, E. and Delp, J. (1999) A Review of Data Hiding in Digital Images. *Proceedings of the Image Processing*, *Image Quality*, *Image Capture Systems Conference*, Savannah, 25-28 April 1999, 274-278.

[35] Kumar, A. and Pooja, K. (2010) Steganography—A Data Hiding Technique. *International Journal of Computer Applications*, **9**, 19-23.

[36] Yadav, D., Agrawal, M. and Arora, A. (2014) Performance Evaluation of LSB and LSD in Steganography. *Proceedings of the* 5*th IEEE International Conference on Confluence the Next Generation Information Technology Summit* (*Confluence*), Noida, 25-26 September 2014, 515-520. https://doi.org/10.1109/confluence.2014.6949380

[37] Artz, D. (2001) Digital Steganography: Hiding Data within Data. *IEEE Internet Computing*, **5**, 75-80. https://doi.org/10.1109/4236.935180

[38] Johnson, N. and Jajodia, S. (1998) Exploring Steganography: Seeing the Unseen. *IEEE Computer*, **31**, 26-34. https://doi.org/10.1109/MC.1998.4655281

[39] Desoky, A. (2012) Noiseless Steganography: The Key to Covert Communications. CRC Press, Boca Raton. https://doi.org/10.1201/b11575

[40] Tiwari, A., Yadav, S. and Mittal, N. (2014) A Review on Different Image Steganography Techniques. *International Journal of Engineering and Innovative Technology*, **3**, 121-124.

[41] Wu, D. and Tsai, W. (2003) A Steganographic Method for Images by Pixel-Value Differencing. *Pattern Recognition Letters*, **24**, 1613-1626.

[42] Chan, C. and Cheng, L. (2004) Hiding Data in Images by Simple LSB Substitution. *Pattern Recognition*, **37**, 469-474. https://doi.org/10.1016/j.patcog.2003.08.007

[43] Wu, C., Wu, I., Tsai, S. and Hwang, S. (2005) Image Steganographic Scheme Based on Pixel-Value Differencing and LSB Replacement Method. *IEE Proceedings*: *Vision*, *Image*, *and Signal Processing*, **152**, 611-615. https://doi.org/10.1049/ip-vis:20059022

[44] Chandramouli, R., Kharrazi, M. and Memon, N. (2004) Image Steganography and Steganalysis: Concepts and Practice. *Proceedings of the* 2*nd International Workshop*, *Book Chapter in Digital Watermarking*, Lecture Notes in Computer Science Series, Seoul, 20-22 October 2003, 35-49. https://doi.org/10.1007/978-3-540-24624-4_3

[45] Martin, A., Sapiro, G. and Seroussi, G. (2005) Is Image Steganography Natural? *IEEE Transactions on Image Processing*, **14**, 2040-2050. https://doi.org/10.1109/TIP.2005.859370

[46] Kamble, V. and Warvante, G. (2013) A Review on Novel Image Steganography Techniques. *IOSR Journal of Computer Engineering*, 1-4.

[47] Rafiuddin, A. and Kumar, C. (2014) Secure Communication with Steganography—An Overview. *International Journal of Recent Research and Review*, **2**, 58-62.

[48] Maleki, N., Jalali, M. and Jahan, V. (2014) Adaptive and Non-Adaptive Data Hiding Methods for Grayscale Images Based on Modulus Function. *Egyptian Informatics Journal*, **15**, 115-127. https://doi.org/10.1016/j.eij.2014.06.001

[49] Chang, C., Chuang, C. and Hu, C. (2006) Spatial Domain Image Hiding Scheme Using Pixel-Values Differencing. *Fundamenta Informaticae*, **70**, 171-184.

[50] El-Alfy, M. and Al-Sadi, A. (2012) High-Capacity Image Steganography Based on Overlapped Pixel Differences and Modulus Function. *Proceedings of the* 4*th International Conference*, Book Chapter in Networked Digital Technologies, Dubai, 24-26 April 2012, 243-252. https://doi.org/10.1007/978-3-642-30567-2_20

[51] El-Alfy, S. and Al-Sadi, A. (2011) A Comparative Study of PVD-Based Schemes for Data Hiding in Digital Images. *Proceedings of the* 9*th IEEE/ACS International Conference on Computer Systems and Applications*, Sharm El-Sheikh, 27-30 December 2011, 144-149. https://doi.org/10.1109/aiccsa.2011.6126588

[52] Khodaei, M. and Faez, K. (2012) New Adaptive Steganographic Method Using Least-Significant-Bit Substitution and Pixel-Value Differencing. *IET Image Processing*, **6**, 677-686. https://doi.org/10.1049/iet-ipr.2011.0059

[53] Wang, M., Wu, I., Tsai, S. and Hwang, S. (2008) A High Quality Steganographic Method with Pixel-Value Differencing and Modulus Function. *Journal of Systems and Software*, **81**, 150-158. https://doi.org/10.1016/j.jss.2007.01.049

[54] Joo, C., Lee, Y. and Lee, K. (2010) Improved Steganographic Method Preserving Pixel-Value Differencing Histogram with Modulus Function. *Journal on Advances in Signal Processing*, 1-13.

[55] Chen, J. (2014) A PVD-Based Data Hiding Method with Histogram Preserving Using Pixel Pair Matching. *Signal Processing*: *Image Communication*, **29**, 375-384. https://doi.org/10.1016/j.image.2014.01.003

[56] Hong, W. and Chen, S. (2012) A Novel Data Embedding Method Using Adaptive Pixel Pair Matching. *IEEE Transactions on Information Forensics and Security*, **7**, 176-184. https://doi.org/10.1109/TIFS.2011.2155062

[57] Lee, F. and Chen, L. (2010) A Novel Data Hiding Scheme Based on Modulus Function. *Journal of Systems and Software*, **83**, 832-843. https://doi.org/10.1016/j.jss.2009.12.018

[58] USC-SIPI Image Database Website. http://sipi.usc.edu/database/database.php?volume=misc

[59] UWATERLOO-LINKS Image Repository Website. http://links.uwaterloo.ca/Repository.html

[60] Gupta, P., Roy, R. and Changder, S. (2014) A Secure Image Steganography Technique with Moderately Higher Significant Bit Embedding. *Proceedings of the* 2014 *IEEE International Conference on Computer Communication and Informatics*, Coimbatore, 3-5 January 2014, 1-6.

[61] Jafar, I.F., Al Na'mneh, R.A. and Darabkh, K.A. (2013) Efficient Improvements on the BDND Filtering Algorithm for the Removal of High-Density Impulse Noise. *IEEE Transactions on Image Processing*, **22**, 1223-1232.

https://doi.org/10.1109/TIP.2012.2228496

[62] Jafar, I., Darabkh, K.A. and Al-Sukkar, G. (2012) A Rule-Based Fuzzy Inference System for Adaptive Image Contrast Enhancement. *The Computer Journal*, **55**, 1041-1057. https://doi.org/10.1093/comjnl/bxr120

[63] Jafar, I. and Darabkh, K.A. (2011) Image Contrast Enhancement Based on Equalization of Edge Histograms. *IAENG International Journal of Computer Science*, **38**, 192-204.

[64] Jafar, I. and Darabkh, K.A. (2011) A Modified Unsharp-Masking Technique for Image Contrast Enhancement. 8*th International Multi-Conference on Systems, Signals and Devices*, Sousse, 22-25 March 2011, 1-6. https://doi.org/10.1109/SSD.2011.5767489

[65] Darabkh, K.A., Awad, A.M. and Khalifeh, A.F. (2015) New Video Discarding Policies for Improving UDP Performance over Wired/Wireless Networks. *International Journal of Network Management*, **25**, 181-202. https://doi.org/10.1002/nem.1888

[66] Darabkh, K.A., Awad, A.M. and Khalifeh, A.F. (2013) Intelligent and Selective Video Frames Discarding Policies for Improving Video Quality over Wired/Wireless Networks. *Proceedings of the* 2013 *IEEE International Symposium on Multimedia*, Anaheim, 9-11 December 2013, 297-300. https://doi.org/10.1109/ISM.2013.57

[67] Darabkh, K.A. and Aygun, R. (2011) Improving UDP Performance Using Intermediate QoD-Aware Hop System for Wired/Wireless Multimedia Communication Systems. *International Journal of Network Management*, **21**, 432-454. https://doi.org/10.1002/nem.768

[68] Darabkh, K.A. and Aygun, R.S. (2006) Performance Evaluation of Sequential Decoding System for UDP-Based Systems for Wireless Multimedia Networks. *Proceedings of* 2006 *International Conference on Wireless Networks*, Las Vegas, 26-29 June 2006, 365-371.

[69] Wang, Z., Bovik, A., Sheikh, H. and Simoncelli, P. (2004) Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, **13**, 600-612. https://doi.org/10.1109/TIP.2003.819861

[70] Darabkh, K.A., Awad, A.M. and Khalifeh, A.F. (2014) Efficient PFD-Based Networking and Buffering Models for Improving Video Quality over Congested Links. *Wireless Personal Communications*, **79**, 293-320. https://doi.org/10.1007/s11277-014-1857-1