A novel pattern-based reversible data hiding technique for video steganography

R. Roselinkiruba (roselinkirubar@ssn.edu.in)
Sri Siva Subramaniya Nadar College of Engineering
T. Sree Sharmila
Sri Siva Subramaniya Nadar College of Engineering
J. K.Josephine Julina
Sri Siva Subramaniya Nadar College of Engineering

Research Article

Keywords: Data hiding, Euclidean Distance, moving object, pattern, weights

Posted Date: May 9th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1619375/v1

A novel pattern-based reversible data hiding technique for video steganography

R. Roselinkiruba *, T. Sree Sharmila, J.K. Josephine Julina

Department of Information Technology, Sri Siva Subramaniya Nadar College of Engineering, Kalavakkam, Chennai, India

roselinkirubar@ssn.edu.in, sreesharmilat@ssn.edu.in, josephinejulinajk@ssn.edu.in

Abstract - This paper explains a novel pattern based color video data hiding technique based on moving regions detection. In the proposed system first the video gets separated into frames, and Discrete Cosine Transform (DCT) compression is applied for every frame. Then, moving objects in the frames are detected and the image is separated into Red, Green, Blue (RGB) channel 2×3 pixel blocks. Further, Euclidean Distance (ED) is used to pair the left over moving pixels into blocks. Based on the hiding capacity of block in each channel, pattern based data hiding process in each channel is implemented. Applying same data hiding methods on all the three channels may affect the quality of the stego image. This can be solved by varying the different data hiding techniques applied on each channels. The proposed Reversible Data Hiding Interpolation with Pixel Value Differencing (RDHI+PVD) method hides the data into the up-scaled image by considering the weights between the pixels. In first channel and second channel, proposed RDHI+PVD and in third channel, only LSB data hiding is applied. Experimental results reveals that the average PSNR, average capacity, RS and PVD histogram analysis attacks obtained on the proposed method performs better than other state-of-the-art techniques.

Keywords— *Data hiding, Euclidean Distance, moving object, pattern, weights*

1. Introduction

Data hiding plays a vital role in information security system carried out by multimedia applications such as text, image, audio, and video to transfer the secret messages [1, 2]. To improve the security against the data breach, high capacity and high image quality are used by the developers [3, 4]. The most common approach of data hiding is the Least Significant Bit (LSB) that hides the secret data inside the LSB bits of the cover image [5]-[8]. However, the LSB replacement technique has resulted in poor image quality when the secret text is increased, which is vulnerable to RS attack [9]. In LSB revisited scheme, the process of hiding carried out by using a pair of pixels, wherein the one bit of information is stored in the first pixel LSB, and two-pixel value functions are stored in another bit. Ni et al. [10] has proposed a reversible data hiding technique based on histogram modification. This algorithm modifies the minimum points of the histogram of an image and slightly improves the pixel values to hide the data inside the image. Zhang and Wang [11] developed a new irreversible data hiding method called Exploiting Modification Direction (EMD), which can withstand RS attack and improved the image quality. The EMD method is based on the secret bit with cover image pixels. Here, data can be embedded by altering one pixel value as +1 or -1. This method has achieved high image quality with no reconstruction of cover image, and it manages less embedding capacity to increase extensive secret data. Chang et al. [12] has developed a reversible data hiding method based on EMD with dual image. Here, two same cover images are chosen for hiding a data. Implementing the data hiding technique on identical images has improved the image quality and has altered the embedding capacity. Many techniques [13, 14] have developed to increase the embedding capacity of EMD.

Steganographic based on a combination of LSB replacement and PVD method is also implemented [15]. The PVD method is used to figure out difference value from two consecutive pixels. In small difference values, the secret text is embedded into the cover image by the LSB method. Khodaei and Faez [16] developed the PVD and LSB substitution for input images. The data is embedded by finding the difference between the center pixel and the nearest pixel. This method lags with the Fall Off Boundary Problem (FOBP). Swain [17] et.al., enhanced the work by dividing the image into 2×2 non-overlapping block. The data is hidden in the upper-left pixel using LSB substitution. Then, the new pixel value is calculated using three different values in both horizontal and vertical directions. This work has extended to hide the data in 2×3 and 3×3 pixel blocks, and finally the data is embedded using LSB substitution [18]. It rectifies three problems FOBP, PDH analysis, and RS analysis. Shu-Yuan, Li-Hong [19] used the cover image to map inside the Hilbert filling curve. Comparing the difference of consecutive pixels with the PVD method the overflow and underflow problems with less distortion of images is solved. Gandharba et. al., [20] proposed two steganography methods, (i) quotient value differencing (QVD), and (ii) pixel value differencing with modulus function (MFPVD). The QVD technique is based on 3×3 size pixel blocks. The data hidden in LSB substitution applied on 2 LSBs, and the remaining 6 bits forms a quotient block. The MFPVD technique uses 2×3 size non-overlapped pixel blocks. By using any one of the pixels as center pixel called as reference value, five neighboring values are generated. The average of these pixels decides the hiding capacity. Furthermore, this method is undetectable by RS analysis and pixel difference histogram (PDH) analysis. The MFPVD method achieves high PSNR, and the QVD method achieves high capacity.

Data-hiding technique using scene-change detection for video Steganography also developed. Mostly, the earlier research in data hiding technique is based on the combination of Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) coefficients. The scene changes are found using the DCT coefficients of video sequences for hiding data. The security and video quality are enhanced using the DWT coefficients technique. It achieves only single layer of data hiding in cover video [21]. Secure video steganographic algorithm in DWT and DCT is based on the Multiple Object Tracking (MOT) algorithm and error correcting codes. The secret message is preprocessed using Hamming and Bose, Chaudhuri, and Hocquenghem codes for encoding the data. The MOT algorithm is used to find the moving objects in the cover video. The data is hidden using DWT and DCT coefficients. This method improves the quality of the image, capacity and withstanding against various attacks. Only DWT and

DCT frequencies are taken for consideration whereas other frequency domains are not considered to improve the stego quality and security [22].

Recent steganography techniques are based on interpolation-based data hiding method. The first technique is the classical bilinear interpolation method which is based on the average of the nearest four pixels. The drawback of this method is it has high computation time, and it mostly works for the smooth image. The second method is the nearest neighbor interpolation method which identifies the closest of the market pixel. This method has the problem with different size of input image (i.e., increase and decreasing the size of image). The third method is the Neighbor Mean Interpolation (NMI) [23] method which finds the mean of the neighboring pixels. It embeds lager amount of data with good image quality. Jung et al. [24] developed Enhanced Neighbor Mean Interpolation (ENMI) that are computed by weighting the nearest neighboring pixels. This method achieves the maximum embedding capacity. Aruna et al. [25] developed the interpolated images based RDH with up-scaled interpolation method with pixel variation range. Data is embedded based on the pixel intensity range. This method suffers in the identification of weighted pixels.

The main contribution of the paper as follows: This paper proposes a secure video steganography method based on compression, moving object identification and pattern-based data hiding. The video is initially separated into frames, and DCT compression is applied to each frame separately. Later, the moving object is identified in the video using AGMM and Kalman filter and then, the image is separated into RGB channel. Then, divide the group of n cover-pixels into 2×3 sub-blocks, each sub-block can embed secret data adaptively inside three channels using Reversible Data Hiding Interpolation with Pixel Value Differencing (RDHI+PVD) method. Using this approach, (i) data hiding inside moving object creates less distortion which make it difficult to detect by the intruder, and also high secrecy for data is achieved by using pattern (ii) it is hard to find the data using PDH security analysis, (iii) embedding data is based on weights on pixel variation which has improved the quality of the stego video (iv) embedding capacity is improved by using three layer of data hiding.

The rest of the paper is organized as follows. The recent literature surveys based on data hiding discussed in section 2. The brief description of the proposed approach explained in Section 3. The results of our experiments are illustrated and compared with other state-of-the-art techniques in Section 4. Finally, in Section 5, the conclusions are drawn, and ideas for the future works are presented.

2. Proposed method

A video steganography for data hiding inside the moving object is illustrated in Fig. 1. The cover video is splitted into frames. Then, the proposed method is applied based on the following steps: (i) video compression (ii) moving object identification and (iii) pattern-based data hiding.

2.1. Video compression

The first step of the proposed data-hiding method is video compression. The most used technique for image compression is the Discrete Cosine Transform (DCT) because of its fast computations and data storage [23, 24]. The cover-video is compressed to remove the spatial and temporal redundancies with the help of the DCT. In video compression, the video is separated into several frames. The DCT compression method changes each image pixel value into the frequency domain. Here, the DCT conversion occurs in such a way that the low frequencies are stored in the top-left and the higher frequencies are stored in the bottom-right. Next, the quantization process occurs, which converts DCT coefficients into an integer that is scaled by the scaling factor. Later, the Inverse Discrete Cosine Transform (IDCT) algorithm is applied to decompress the original input frames.

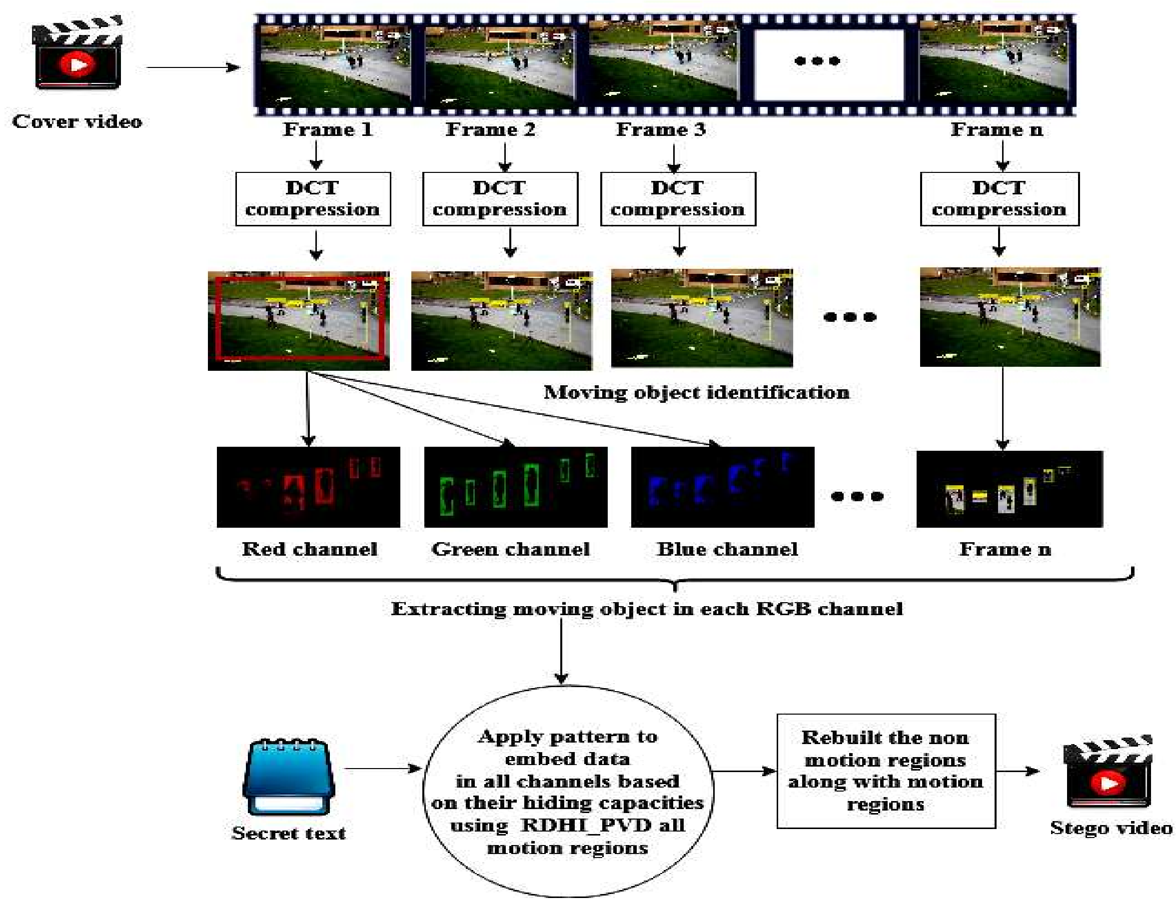


Fig. 1 Block diagram of the proposed RDHI+PVD technique for video steganography

2.2. Moving object identification

After compression, the next task is to identify the moving objects in the frame. In order to detect the moving object, a small dynamic object in the video frame is taken into consideration. The detection of a moving object is done by two stages: 1) identification of moving object in every individual video frame 2) tracking of the moving object in all the video frames.

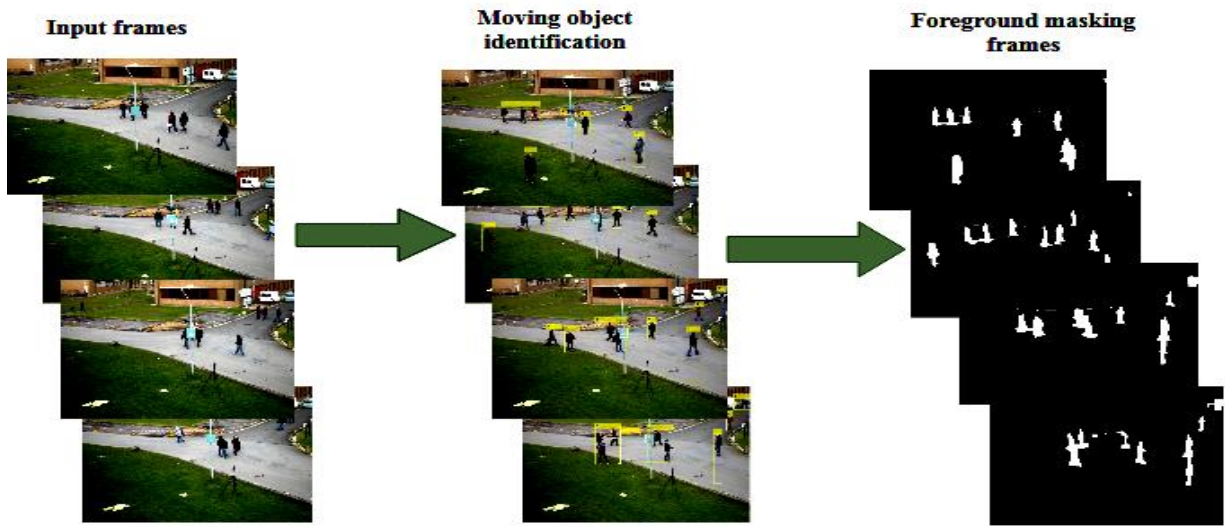


Fig. 2 Left column: four video frames from S2L1 PETS2009 dataset [28], middle column: finding numerous moving objects between the frames, and right column: foreground masks frames.

Here, the Regions-Of-Interest (ROI) of the moving objects are identified using Adaptive Gaussian Mixture Model (AGMM). The primary purpose of AGMM is to separate the background and foreground masks. In proposed AGMM, the morphological operations are used to focus more on small dynamic moving objects. The background difference between two successive frames that gives the foreground mask is shown in Fig. 2. The morphological functions remove all the noise in the foreground mask. As a result, the moving objects are sorted from groups of significant pixels. In the second stage, the object's motion is detected using the Kalman Filter (KF) [29, 30]. The KF predicts the current position of each object. Additionally, KF is determined to handle the probability of detection of moving object in each path. The AGMM method and the Kalman filter can detect the occlusion object and number of moving objects.

2.3. Pattern-based video data hiding using RDHI and PVD method

To provide both higher payload capacity and better stego-image quality this paper proposes a novel pattern-based data hiding technique using RDH interpolation and PVD. In data hiding stage, the input video is separated into frames, and DCT compression applied. A moving object in each frame is found by differencing the background between two successive frames. The background is extracted using the AGMM method, and moving objects are predicted using KF. Then, background values are extracted and are shown in black color (zero value pixels), and the moving object is separated, and represented in white color pixel (one value pixels). The white pixels or

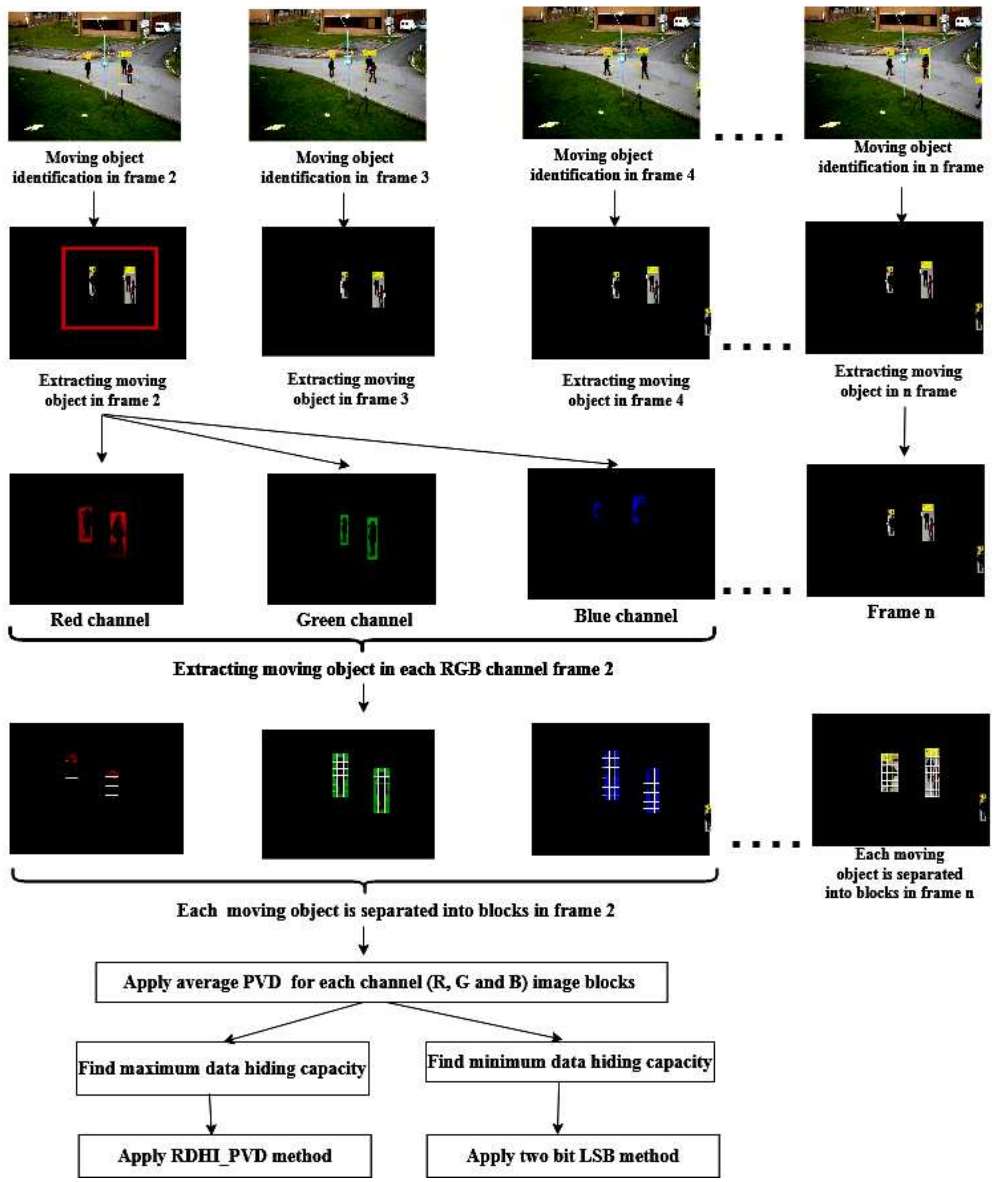
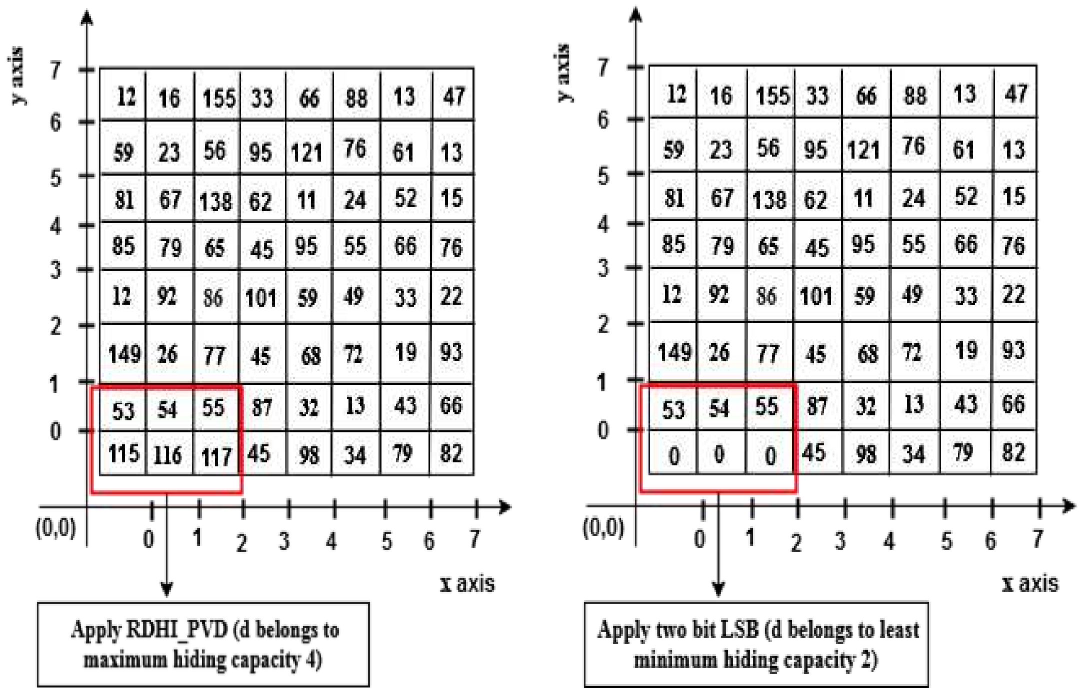


Fig. 3 Moving object identification and block selection process using RDHI+PVD algorithm

foreground masked frames are multiplied with original frame to get the extracted moving object frame. Here, the moving object frames are separated into 2×3 blocks. The pattern is applied for each moving object for embedding data. The three layers of data hiding techniques followed in the proposed method and are carried out in the RGB channel. In first and second channel, data is hidden in all the moving regions using RDH interpolation method. In the third channel, data is hidden inside the LSB of the next channel. The channel is chosen based on their capacity using average PVD using Eq. (4).

Each channel blocks data hiding capacity is found using the range Table 1. Here, sample 8×8= 64 pixels are taken into consideration. For understanding, the pixels are plotted in graph with x-axis and y-axis representation. Consider, 2×3 block with the range *D[64, 255]* and its hiding capacity *t=4* is depicted in Fig. 4(a). Similarly, Figure 4(b) shows 2×3 block with the range *D[0,30]* and its hiding capacity *t=2*. The data hiding capacity (t) channels in each block are selected and the ranges are illustrated in Table 2. In maximum capacity and in the next maximum capacity channel the RDHI and PVD method is applied and in the last channel two-bit LSB method is applied. In case two pixels contain same maximum data hiding. Example: (i) If R and G has the same maximum data capacity the R is chosen, (ii) If G and B has the same maximum data capacity the G is chosen and (iii) If G and B has the same maximum data capacity the G is chosen. The procedure is same for next maximum hiding capacity. The detailed explanation of the block selection process using RDH interpolation algorithm explained in Fig. 3 and Fig 4.



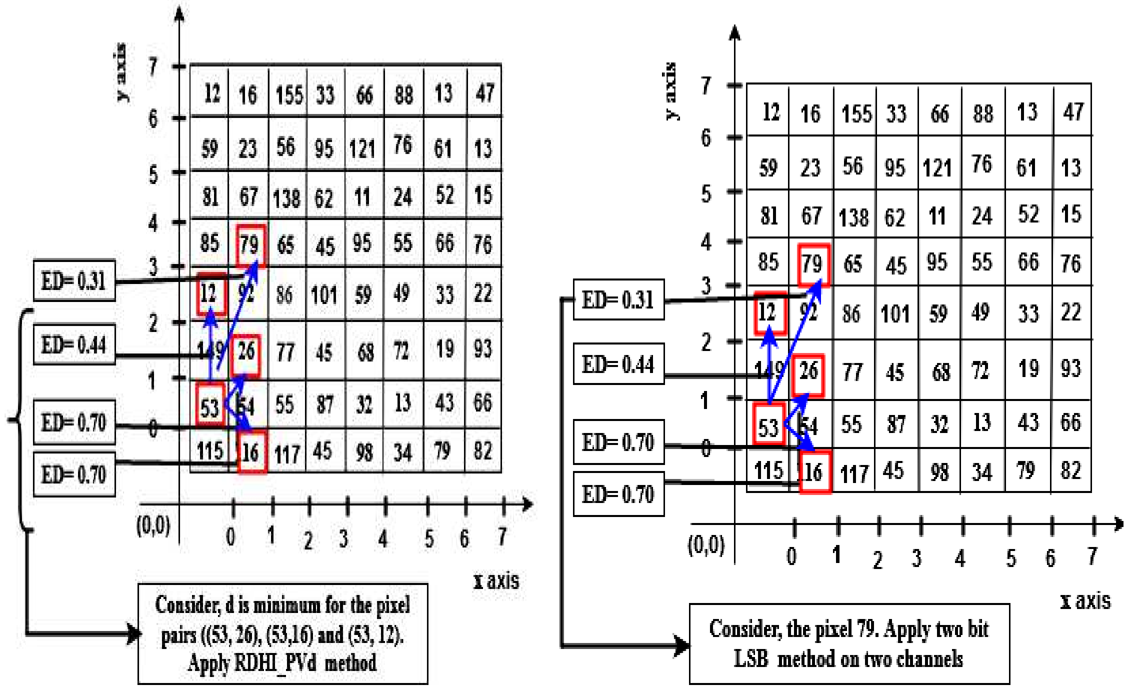
(a) **(b)**

Fig. 4 Block identification process in channel using RDHI and PVD algorithm (a) assume maximum hiding capacity (b) assume leas hiding capacity

2.3.1 Distance calculation

After separation of 2×3 blocks, the remaining 1×1 blocks are paired with the nearest moving region pixel and form a group of pixels with the block of minimum 1×3 pixels. The distance between these two pixels is calculated using Equation (1).

$$ED = \frac{1}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \tag{1}$$



(a) **(b)**

Fig. 5 Pixel pair identification (a) pixel with minimum ED range (b) pixel with maximum ED range

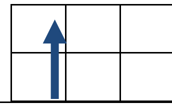
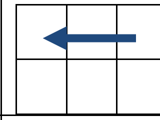
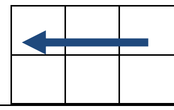
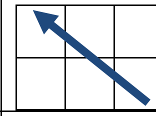
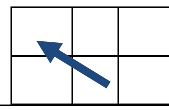
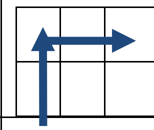
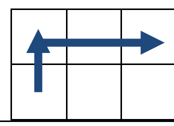
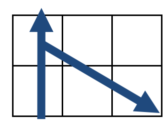
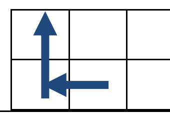
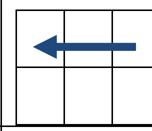
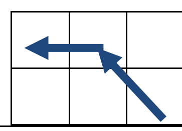
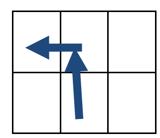
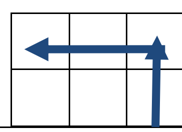
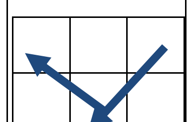
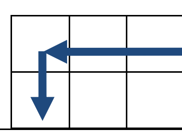
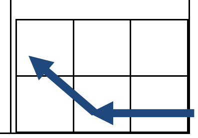
Consider the pixels = (53, 26, 12, 79) in the x and y axis. Find the nearest pixel of pixel 53 using ED is illustrated in Figure 5. The distance of pixel (53, 26) is 0.70, (53, 16) is 0.70, (53, 13) is 0.44 and (53, 79) is 0.31. The shortest distances between the pixels in the range is chosen then to apply RDHI and PVD technique. If the pixel pair is not in the range then apply 2-bit LSB method on three channels on the pixel 79.

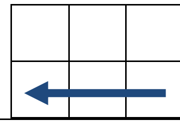
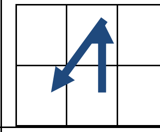
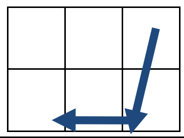
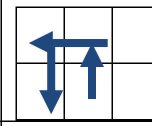
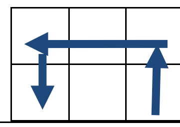
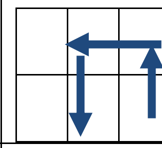
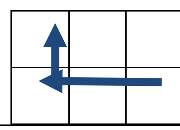
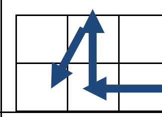
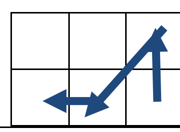
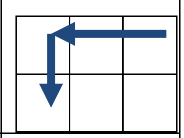
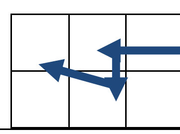
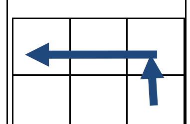
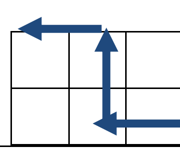
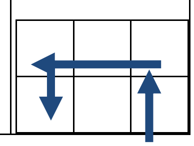
2.3.2 Pattern for moving objects

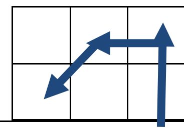
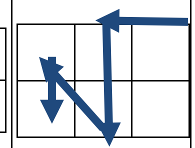
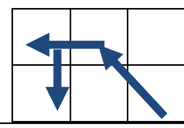
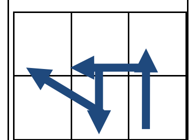
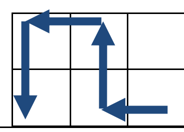
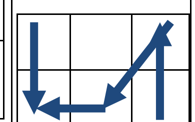
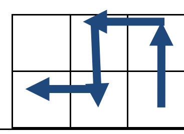
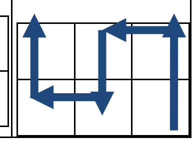
After the blocks formation using the ED, the pattern is applied for each block. If there is a single moving object in blocks no need to apply pattern for that blocks. The detailed explanation of each block pattern is explained in Table 1. Where, MO represents the moving object.

Table 1. Pattern for moving objects

Rule	Diagram representation	Pattern output	Rule	Diagram representation	Pattern output

<table>
<tr><td>1</td><td><table><tr><td>MO</td><td></td><td></td></tr><tr><td>MO</td><td></td><td></td></tr></table></td><td></td><td>2</td><td><table><tr><td>MO</td><td>MO</td><td></td></tr><tr><td></td><td></td><td></td></tr></table></td><td></td></tr>
<tr><td>3</td><td><table><tr><td>MO</td><td></td><td>MO</td></tr><tr><td></td><td></td><td></td></tr></table></td><td></td><td>4</td><td><table><tr><td>MO</td><td></td><td></td></tr><tr><td></td><td></td><td>MO</td></tr></table></td><td></td></tr>
<tr><td>5</td><td><table><tr><td>MO</td><td></td><td></td></tr><tr><td></td><td>M
O</td><td></td></tr></table></td><td></td><td>6</td><td><table><tr><td>MO</td><td></td><td>MO</td></tr><tr><td>MO</td><td></td><td></td></tr></table></td><td></td></tr>
<tr><td>7</td><td><table><tr><td>MO</td><td></td><td>MO</td></tr><tr><td>MO</td><td></td><td></td></tr></table></td><td></td><td>8</td><td><table><tr><td>MO</td><td></td><td></td></tr><tr><td>MO</td><td></td><td>MO</td></tr></table></td><td></td></tr>
<tr><td>9</td><td><table><tr><td>MO</td><td></td><td></td></tr><tr><td>MO</td><td>MO</td><td></td></tr></table></td><td></td><td>10</td><td><table><tr><td>MO</td><td>MO</td><td>M
O</td></tr><tr><td></td><td></td><td></td></tr></table></td><td></td></tr>
<tr><td>11</td><td><table><tr><td>MO</td><td>MO</td><td></td></tr><tr><td></td><td></td><td>MO</td></tr></table></td><td></td><td>12</td><td><table><tr><td>MO</td><td>MO</td><td></td></tr><tr><td></td><td>MO</td><td></td></tr></table></td><td></td></tr>
<tr><td>13</td><td><table><tr><td>MO</td><td></td><td>MO</td></tr><tr><td></td><td></td><td>MO</td></tr></table></td><td></td><td>14</td><td><table><tr><td>MO</td><td></td><td>MO</td></tr><tr><td></td><td>M
O</td><td></td></tr></table></td><td></td></tr>
<tr><td>15</td><td><table><tr><td>MO</td><td>M
O</td><td></td></tr><tr><td>MO</td><td></td><td></td></tr></table></td><td></td><td>16</td><td><table><tr><td>M
O</td><td></td><td></td></tr><tr><td></td><td>MO</td><td>MO</td></tr></table></td><td></td></tr>
</table>

<table>
<tr><td>17</td><td><table><tr><td></td><td></td><td></td></tr><tr><td>MO</td><td>MO</td><td>MO</td></tr></table></td><td></td><td>18</td><td><table><tr><td></td><td>MO</td><td></td></tr><tr><td>MO</td><td>MO</td><td></td></tr></table></td><td></td></tr>
<tr><td>19</td><td><table><tr><td></td><td></td><td>MO</td></tr><tr><td>MO</td><td>MO</td><td></td></tr></table></td><td></td><td>20</td><td><table><tr><td>MO</td><td>MO</td><td></td></tr><tr><td>MO</td><td>MO</td><td></td></tr></table></td><td></td></tr>
<tr><td>21</td><td><table><tr><td>MO</td><td></td><td>MO</td></tr><tr><td>MO</td><td></td><td>MO</td></tr></table></td><td></td><td>22</td><td><table><tr><td></td><td>MO</td><td>MO</td></tr><tr><td></td><td>MO</td><td>MO</td></tr></table></td><td></td></tr>
<tr><td>23</td><td><table><tr><td>MO</td><td></td><td></td></tr><tr><td>MO</td><td>MO</td><td>MO</td></tr></table></td><td></td><td>24</td><td><table><tr><td></td><td>MO</td><td></td></tr><tr><td>MO</td><td>MO</td><td>MO</td></tr></table></td><td></td></tr>
<tr><td>25</td><td><table><tr><td></td><td></td><td>MO</td></tr><tr><td>MO</td><td>MO</td><td>MO</td></tr></table></td><td></td><td>26</td><td><table><tr><td>MO</td><td>MO</td><td>MO</td></tr><tr><td>MO</td><td></td><td></td></tr></table></td><td></td></tr>
<tr><td>27</td><td><table><tr><td>MO</td><td>MO</td><td>MO</td></tr><tr><td></td><td>MO</td><td></td></tr></table></td><td></td><td>28</td><td><table><tr><td>MO</td><td>MO</td><td>MO</td></tr><tr><td></td><td></td><td>MO</td></tr></table></td><td></td></tr>
<tr><td>29</td><td><table><tr><td>MO</td><td>MO</td><td></td></tr><tr><td></td><td>MO</td><td>MO</td></tr></table></td><td></td><td>30</td><td><table><tr><td>MO</td><td>MO</td><td></td></tr><tr><td>MO</td><td></td><td>MO</td></tr></table></td><td></td></tr>
</table>

<table>
<tr><td>31</td><td><table><tr><td></td><td>MO</td><td>MO</td></tr><tr><td>MO</td><td></td><td>MO</td></tr></table></td><td></td><td>32</td><td><table><tr><td>MO</td><td>MO</td><td>MO</td></tr><tr><td>MO</td><td>MO</td><td></td></tr></table></td><td></td></tr>
<tr><td>33</td><td><table><tr><td>MO</td><td>MO</td><td>MO</td></tr><tr><td>MO</td><td></td><td>MO</td></tr></table></td><td></td><td>34</td><td><table><tr><td>MO</td><td>M
O</td><td>MO</td></tr><tr><td></td><td>M
O</td><td>MO</td></tr></table></td><td></td></tr>
<tr><td>35</td><td><table><tr><td>MO</td><td>MO</td><td></td></tr><tr><td>MO</td><td>MO</td><td>MO</td></tr></table></td><td></td><td>36</td><td><table><tr><td>MO</td><td></td><td>MO</td></tr><tr><td>MO</td><td>M
O</td><td>MO</td></tr></table></td><td></td></tr>
<tr><td>37</td><td><table><tr><td></td><td>MO</td><td>MO</td></tr><tr><td>MO</td><td>MO</td><td>MO</td></tr></table></td><td></td><td>38</td><td><table><tr><td>MO</td><td>MO</td><td>MO</td></tr><tr><td>MO</td><td>MO</td><td>MO</td></tr></table></td><td></td></tr>
</table>

2.3.3 Data embedding

The data is embedded in the pattern fashion and the image is divided into non-overlapping 2×3 pixel blocks. If the block contains only one pixel as moving object, 4-bit LSB method is used for hiding the data inside the pixel. The data hiding within the blocks are done by up-scaling each block. The interpolation-based data hiding calculates the weight of each pixel by using Pixel Value Differencing (PVD). The weight of a particular pixel is estimated as the difference between the pixel value of the current and the neighboring pixel. The detailed description of the algorithm is given below:

1) Estimate Pixel Value Difference (PVD) between the two pixels using Eq. (4).

$$D = \left[\frac{1}{n} \sum_{i=1}^{n} I_x - I_i \right] \tag{4}$$

Where, n is the total number of pixels. For the finding single pixel pair, the $\frac{1}{n}$ value is neglected

Table 2. Description of range table with weights

PVD difference	Weight	Actual hiding capacity (t)
Input image	W=0	2
[0-30]	W=1	3
[31-65]	W=2	4
[65-225]	W=3	5

2) The secret text converted into binary bits and the decimal values of binary bits are embedded. The range table with weights is displayed in Table 2.
3) Find C(i,j) based on the input image *I(i,j)* with the size of *N*×*N*.

(i) If *(W=0)* then the *C(i,j)* is calculated as follows:

$$C(i,j) = \begin{cases} I(i,j) \;\; if\ i\ mod2 = 0,\ j\ mod\ 2 = \ 0\ and\ j < N-1 \\ I(i,j) \quad\quad if\ i\ mod2 = 0, and\ j < N-1 \\ I(i,j) \quad\quad if\ j\ mod2 = 0, and\ i < N-1 \end{cases} \tag{5}$$

(ii) If (W=1) then C(i,j) is calculated as follow:

$$C(i,j) = \begin{cases} \frac{I(i-1,j)*2+I(i+1,j)*2+I(i-1,j-2)+I(i+1,j-2)}{6} if\ i\ mod2 = 1\ and\ j = N-1 \\ \frac{I(i,j-1)*2+I(i,j+1)*2+I(i-2,j-1)+I(i-2,j+1)}{6} if\ j\ mod2 = 1\ and\ i = N-1 \\ \frac{I(i,j-1)*2+I(i,j+1)*2+I(i+2,j-1)+I(i+2,j+1)}{6} if\ i\ mod2 = 0, j\ mod2 = 1\ and\ i < N-1 \\ \frac{I(i-1,j)*2+I(i+1,j)*2+I(i-1,j+2)+I(i+1,j+2)}{6} if\ i\ mod2 = 1, j\ mod2 = 0\ and\ j < N-1 \\ \frac{I(i-1,j-1)+I(i-1,j+1)+I(i+1,j-1)+I(i+1,j+1)}{4} otherwise \end{cases} \tag{6}$$

(ii) If *(W=2)* then the *C(i,j)* is calculated as follows:

$$C(i,j) = \begin{cases} \frac{I(i-1,j)+I(i+1,j)+I(i-1,j-2)+I(i+1,j-2)}{4} if\ i\ mod2 = 1\ and\ j = N-1 \\ \frac{I(i,j-1)+I(i,j+1)+I(i-2,j-1)+I(i-2,j+1)}{4} if\ j\ mod2 = 1\ and\ i = N-1 \\ \frac{I(i,j-1)+I(i,j+1)+I(i+2,j-1)+I(i+2,j+1)}{4} if\ i\ mod2 = 0, j\ mod2 = 1\ and\ i < N-1 \\ \frac{I(i-1,j)+I(i+1,j)+I(i-1,j+2)+I(i+1,j+2)}{4} if\ i\ mod2 = 1, j\ mod2 = 0\ and\ j < N-1 \\ \frac{I(i-1,j-1)+I(i-1,j+1)+I(i+1,j-1)+I(i+1,j+1)}{2} otherwise \end{cases} \tag{7}$$

(ii) If *(W=3)* then the *C(i,j)* is calculated as follows:

$$C(i,j) = \begin{cases} \frac{I(i-1,j)+I(i+1,j)+I(i-1,j-2)+I(i+1,j-2)}{2} if\ i\ mod2 = 1\ and\ j = N-1 \\ \frac{I(i,j-1)+I(i,j+1)+I(i-2,j-1)+I(i-2,j+1)}{2} if\ j\ mod2 = 1\ and\ i = N-1 \\ \frac{I(i,j-1)+I(i,j+1)+I(i+2,j-1)+I(i+2,j+1)}{2} if\ i\ mod2 = 0, j\ mod2 = 1\ and\ i < N-1 \\ \frac{I(i-1,j)+I(i+1,j)+I(i-1,j+2)+I(i+1,j+2)}{2} if\ i\ mod2 = 1, j\ mod2 = 0\ and\ j < N-1 \\ \frac{I(i-1,j-1)+I(i-1,j+1)+I(i+1,j-1)+I(i+1,j+1)}{1} otherwise \end{cases} \quad (8)$$

4) The data embedding capacity is decided based on weights which is displayed in Table 2.

Algorithm 1: Proposed algorithm

Input: Cover Video (CV), Secret Text (ST)

Output: Stego Video (SV)

1. Read the cover video CV and partition CV= $\{f_{1,} f_{2,} .....f_n\}$, n denotes the total number of frames.
2. Apply DCT Compression for $\{f_{1,} f_{2,} .....f_n\}$;
3. Find moving object in frame $\{f_{1,} f_{2,} .....f_n\}$;

for k=1 to f_n **do**
{
Identify the MO for each video frame f
Track the ROI (moving object) separately using the foreground subtraction method
for each frame (f) in MO **do**
{
ROI is extracted in all frames (f)
The morphological operation for moving object is separated
}
}

4. Frame $\{f_{1,} f_{2,} .....f_n\}$ are separated into channel C= $\{R_, G_, B\}$ image

for k=1 to f_n **do**
{
Divide the RGB channel image into 2×3 pixel blocks
Compute PVD and hiding capacity t for each channel using Eq. (4)

for (i = 1 : P) **do** // P represents pixels in 2×3 blocks
{
if (t= C (maximum)) **then**
{
// First and second layer data hiding : RDHI+PVD method in two maximum data hiding capacity channel
Find PVD for the two neighboring pixels
Compute weight for the pixels
Find the up scaling of the interpolated pixel values
}

ST is converted into DV, DV denotes Decimal Value;
Embed data based on the weight and data hiding capacity
}
// Third layer data hiding: LSB method in next channel
else (t= C (minimum))
{

```
                    Implement 2-bit LSB data hiding method in all up-scaling values
                    }

   for ( i = 1 : P ) do  // P represents pixels in 1×1 blocks
     {
                    Apply Euclidean Distance (ED) for pixels
       if (ED= range) then
           {
                    Implement (RDHI+PVD) and LSB method in three channels for minimum 1×3 blocks
         else
           {
                    Implement 2-bit LSB method in three channel

                }}
       }
     }

5.   return SV;
```

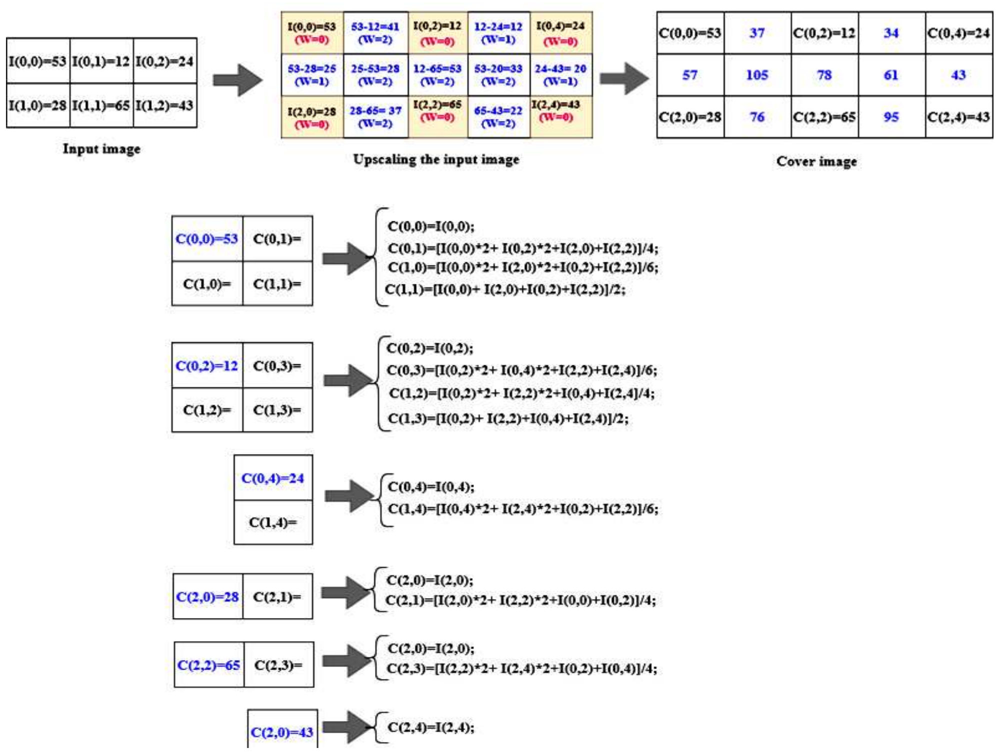


Fig.6. Example of weight-based neighbor mean interpolation method

Let us consider a sample 2×3 sub-block with the pixel values 53, 12, 24, 28, 65 and 43. The interpolation is applied for the input image I*(i, j).* After the interpolation, the output image is up scaled with the size of 5×3 sub-block. The interpolation method is applied by calculating weights between the pixels using PVD. The weight of each block is based on the intensity of the pixel variation shown in the up-scaled input image. The data is hidden inside the cover image. Consider the secret data= 1010101011101 10100100101100101 1010010110110 and it is hidden inside all the pixels. In this method, number of data hidden inside each pixel is based on weight which is illustrated in Table 2. Consider the pixel 53 whose weight *(W=0)* and its binary value is 00110101; hide 2-bit of secret data (10) in the LSB of the pixel. After replacing the secret data, the pixel value will be 00110110= 54. The next pixel is 41 whose weight *(W=2)* and its binary value is 00001010; hide 4-bit of secret data (1010) in the LSB of the pixel. After replacing the secret data, the pixel value will be 00101010= 42. The next pixel is 12 whose weight *(W=0)* and its binary value is 00001100; hide 2-bit of secret data (10) in the LSB of the pixel. After replacing the secret data, the pixel value will be 00001110= 14. The next pixel is 34 whose weight *(W=1)* and its binary value is 00100010; hide 3-bit of secret data (111) in the LSB of the pixel. After replacing the secret data, the pixel value will be 00100111= 39. This procedure is repeated for all the pixels in the blocks and all the interpolated pixels are shown in the Figure 6.

2.3.4 Data extraction

The following steps illustrates how the data is extracted in the proposed method:

(a) The stego video is separated into frames.
(b) Apply inverse DCT compression in each frame. Identify the moving regions using AGMM method and the Kalman filter. The moving object in the image is divided into RGB channel. The data hiding capacity channels are found in each 2×3 or 2×2 or 1×3 pixel blocks using Eq. (4).
(c) Find ED for the single pixel and group them into blocks according to their shortest distance. If the pixels blocks do not fit in the range, which are less than 1×3 then apply 2-bit LSB in three channels. After identification of blocks, the data are extracted in the pattern fashion.
(d) Find the interpolated pixels in the stego image in each channel.
(e) Based on the PVD range, extract the data, and finally remove the interpolated pixels to get the original image.

Let us consider a sample stego block along with the interpolated pixels is given for extracting secret data as shown in the Fig. 7. The pixel values of the stego images are 54, 42, 14, 15, 25, 61, 105, 66, 60, 45, 30, 68, 66, 90, and 42. The extraction of secret data is continued for all the image pixels as shown in the example. Finally, the complete extracted secret bit is 1010101011101 10100100101100101 1010010110110.

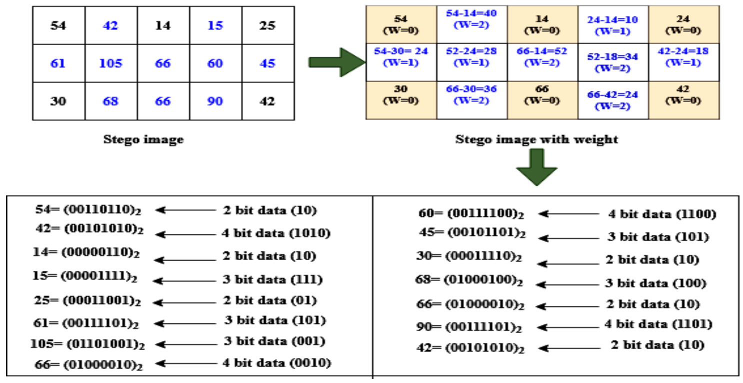


Fig.7. Secret data extraction

3. Experimental results and discussions

This paper shows the capability of the proposed video steganography in tracking the object and hiding the data experimentally. The sample input text of 5 KB, 10 KB and 20 KB were employed to evaluate the performance of the proposed method. These experiments were executed in the MATLAB 15b platform with Intel core I5, 2.7 GHz and 4GB memory PC. The experimental outcome demonstrates that the proposed approach can improve the quality of image, capacity and secure against attacks.

3.1. Dataset

The proposed multiple objects tracking method is tested using PETS 2009 database [28]. Table 3 illustrates a brief explanation of all the input video. The test database contains multiple objects moving inside and outside the scene. The entire moving object is tracked in all the video sequences using AGMM and KF, which improves the number of moving object.

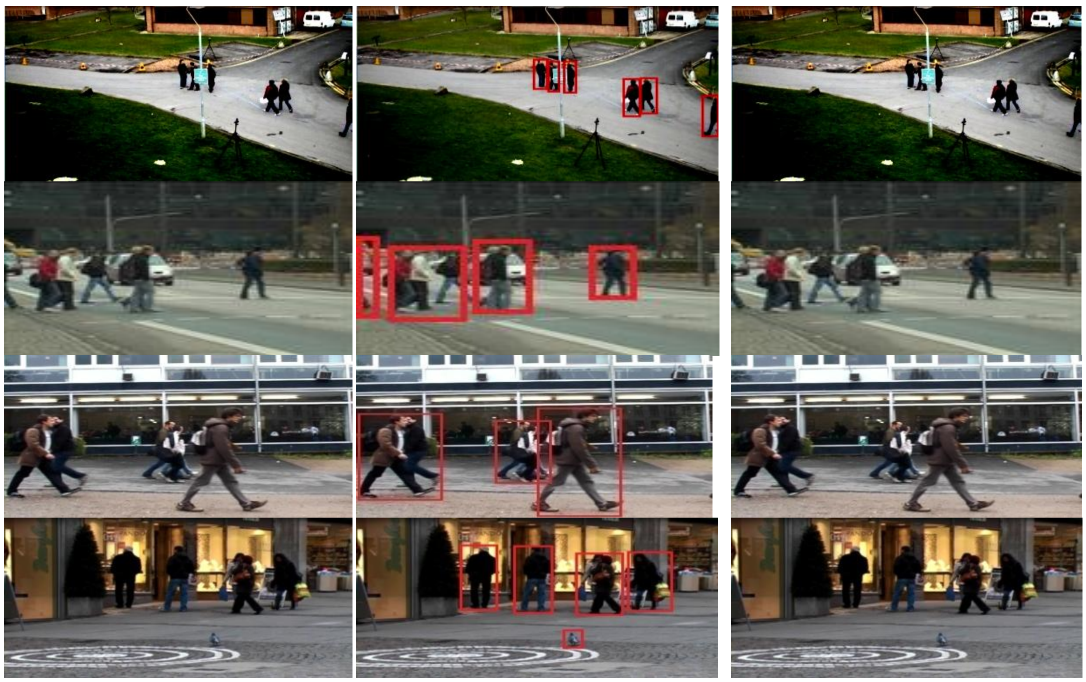
Table 3. Description of the cover video

S.No	Input video	Video size	Resolution	Frames per second
1.	PETS09-S2L1.avi	89,37,150 bytes	768x576	7
2.	TUD-Crossing.avi	71,10,000 bytes	640x480	25
3.	TUD-Campus.avi	11,88,139 bytes	640x480	25
4.	TUD-Stadmitte.avi	29,11,307 bytes	640x480	25

3.2. Performance metrics

The performance of the proposed methodology is measured by using four input cover videos PETS09-S2L1, TUD-Crossing, TUD-Campus and TUD-Stadtmitte. Their corresponding stego videos are displayed in Fig. 8. It is observed that Fig. 8 (a) and (c) that the original and the stego frame does not show much variation, which is not easily detectable by human vision. The advantage of hiding data inside a moving object of the video decreases the chances of visual identification by human and the pattern-based data hiding improves the security. In this paper, the moving object is identified by AGMM and KF method and this is clearly illustrated in Fig. 8. Here, the improved RDHI and PVD technique is used to hide the data inside the cover video. The performance metrics, PSNR [31, 32] and capacity [33, 34] are used to find the stego-videos quality. Then, their resistance against attacks is measured using PVD histogram analysis [35] and RS analysis [36].

(a)
(b)
(c)
(d)



Sample frame for each video **Moving object identification** **Stego frame for each video**

Fig.8. Cover video frame (a) PETS09-S2L1, (b) TUD-Crossing, (c) TUD-Campus and (d) TUD-Stadtmitte.

Subsequently, the experimental results of various cover videos are discussed and evaluated using three metrics namely: (i) PSNR, (ii) embedding capacity and (iii) processing time. The Peak Signal-to-Noise Ratio (PSNR) [33] is used to find the numerical difference between the input video and the output video in decibels (dB). It is computed using Eq. (9). In Eq. (9), L represents the maximum signal position (i.e., 255). The Mean Squared Error (MSE) [31] finds the average square of errors, which computes the variation between the cover and distorted videos using Eq. (10).

$$PSNR = 10\log_{10}\frac{L^2}{MSE} \qquad (9)$$

$$MSE = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(A(i,j) - P(i,j)\right)^2 \qquad (10)$$

Where, H×W is the frame size and A(i, j) is the intensity value of the pixel located at the coordinates (i, j) in the original frame, and P(i, j) is the same pixel in the stego-frame.

The capacity [34] is defined as the maximum amount of embedded information hidden inside the original video which is illustrated in Eq. (11).

$$Capacity = \frac{E}{I} \tag{11}$$

Where E is the highest hidden information, and I is the maximum video size.

Table 4. Performance of PSNR for the RDHI and PVD method in different channel

Input video	**Video size**	**Total number of embedded bits**	**PSNR (RDHI and PVD (in 3 channel))**	**PSNR (RDHI and PVD (in 2 channel), and one channel LSB)**
PETS09-S2L1 (V1)	89,37,150 bytes	8,096 bytes (D1)	42.21	**44.57**
TUD-Crossing (V2)	71,10,000 bytes	8,096 bytes (D1)	41.55	43.26
TUD-Campus (V3)	11,88,139 bytes	8,096 bytes (D1)	41,84	44.01
TUD- Stadtmitte (V4)	29,11,307 bytes	8,096 bytes (D1)	42.01	**44.32**
PETS09-S2L1 (V1)	89,37,150 bytes	12,288 bytes (D2)	42.98	**43.83**
TUD-Crossing (V2)	71,10,000 bytes	12,288 bytes (D2)	41.06	42.69
TUD-Campus (V3)	11,88,139 bytes	12,288 bytes (D2)	40.12	42.32
TUD- Stadtmitte (V4)	29,11,307 bytes	12,288 bytes (D2)	42.63	**43.85**
PETS09-S2L1 (V1)	89,37,150 bytes	36,864 bytes (D3)	37.31	**39.05**
TUD-Crossing (V2)	71,10,000 bytes	36,864 bytes (D3)	36.30	38.74
TUD-Campus (V3)	11,88,139 bytes	36,864 bytes (D3)	36.59	38.78
TUD- Stadtmitte (V4)	29,11,307 bytes	36,864 bytes (D3)	36.67	**38.95**

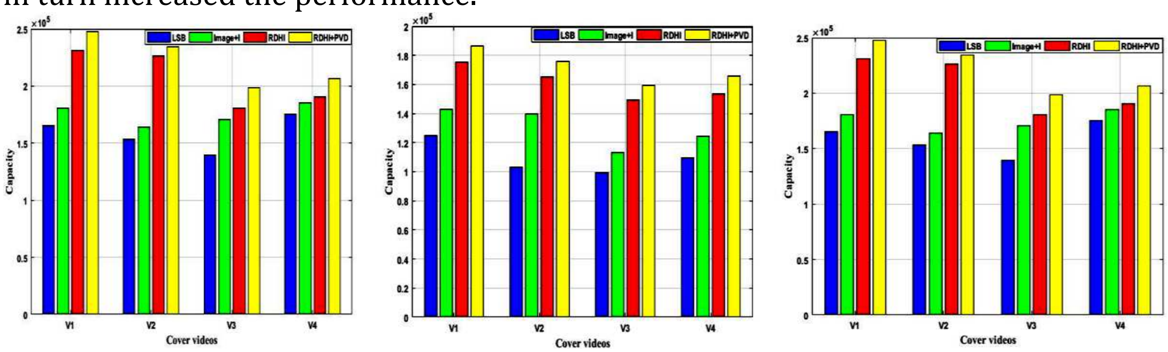
Table 4 shows that the PSNR of the proposed method. Data hiding using RDHI and PVD method in all three channels decreases the PSNR of the proposed method. Hence, we have choosen two-layer data hiding using RDHI and PVD method in the last channel LSB.

Table 5. Performance of PSNR for the existing and the proposed method

Input video	**Total number of embedded bits**	**PSNR (dB)**			
		LSB [37]	Image+I [38]	RDHI [24]	RDHI+PVD
V1	8,096 bytes (D1)	35.56	37.44	43.37	**44.57**
V2	8,096 bytes (D1)	33.97	36.90	42.43	43.26
V3	8,096 bytes (D1)	34.13	37.56	43.77	44.01
V4	8,096 bytes (D1)	35.77	37.23	43.54	**44.32**
V1	12,288 bytes (D2)	34.87	36.57	42.45	**43.83**
V2	12,288 bytes (D2)	32.67	35.75	41.99	42.69
V3	12,288 bytes (D2)	33.28	36.67	42.55	42.32

V4	12,288 bytes (D2)	34.87	36.40	42.89	**43.85**
V1	36,864 bytes (D3)	30.05	32.04	37.34	**39.05**
V2	36,864 bytes (D3)	30.92	31.23	37.16	38.74
V3	36,864 bytes (D3)	31.17	34.54	37.87	38.78
V4	36,864 bytes (D3)	31.62	33.39	36.77	**38.95**

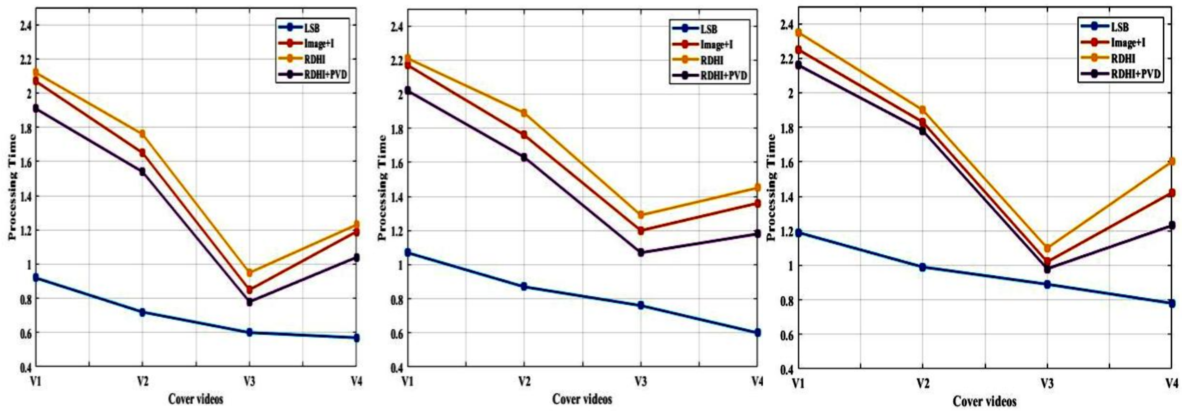
From Table 5, it is clearly observed that the proposed RDHI and PVD method performs well in all input video and are above 37dB which are higher than LSB [32], Image+I [33], and RDHI [34] respectively. The average PSNR of the proposed RDHI and PVD method is 41.79 dB. Similarly, PSNR for LSB, Image Interpolation (Image+I) and Reversible Data Hiding Interpolation (RDHI) are 33.24 dB, 35.47 dB and 42.03 dB, respectively. The LSB method has the low performance while compared to other methods with the advanced interpolation. In Image+I and RDHI technique, the interpolation is calculated by the mean value and distance. It does not concentrate the pixel variation of each pixel which degrades the quality of the image. Whereas, the proposed method takes weight of the each pixels and adopts the range table which in turn increased the performance.



(a) (b) (c)

Fig.9. Performance metrics of capacity with cover videos: a) Hiding data in one channel, (b) Hiding data in two channels and (c) Hiding data in three channels

The performance metrics, capacity for four input cover videos is also implemented. In LSB [35] algorithm and Image+I [36], data is hidden inside one channel without interpolation that reduces the capacity of the data hiding. Further, in RDHI [37], the data hiding takes place only in the up-scaled image which reduces the average capacity of the video. The analysis of Fig. 9 helps to establish that the hiding capacity of the proposed technique is higher than that of the three existing techniques. The hiding capacity is based on the weight of all pixels inside the frame. The three layers of hiding capacity is achieved in the proposed RDHI and PVD inside two channels, which improves the hiding capacity. In RDHI and PVD, the data is hidden based on the range table it varies from 2, 3 and 4. In LSB, two-bit data is achieved uniformly in the moving object pixels. In addition, the DCT compression algorithm is used to improve the storage capacity of frame. The hiding capacity for proposed RDHI+PVD is 164,837 bits, which is better compared to LSB, Image+I and RDHI. Further, the proposed RDHI+PVD method hides the data inside the entire cover image and the interpolated pixels. The above comparison outcome reveals that the performance of the proposed method is higher than that of the existing method following embedding capacity.



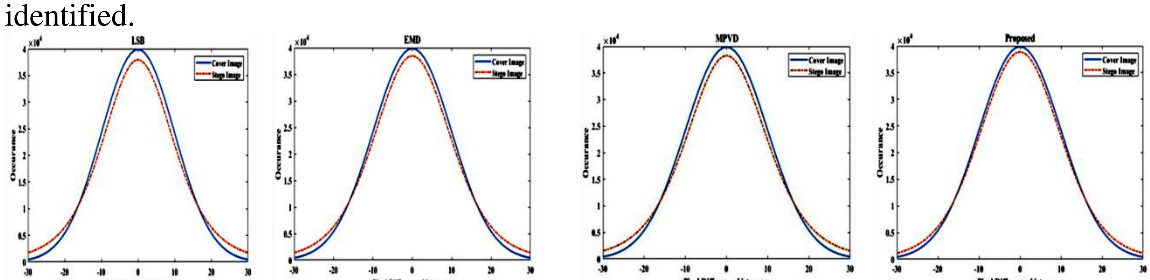
(a) (b) (c)

Fig.10. Processing time (a) 5 KB text, (b) 10 KB text and (c) 20 KB text.

The data embedding and extracting time of the proposed work is analysed with the state-of-the-art algorithm which are illustrated in Fig.10. The processing time of the LSB algorithm is quite low when compared to the other existing methods such as Image+I and RDHI. In LSB algorithm, it hides data directly inside the least significant bit of the moving object. Other algorithms such as Image+I, RDHI and RDHI+PVD compare the range table for hiding data. In Image+I and RDHI, grouping of the partner pixel identification and finding correct partner in a group consumes more time when compared to the other methods. The data is hidden directly in the moving object pixels using RDHI+PVD which reduces the pixel selection process.

3.3. Security analysis

The security of the proposed steganography technique is evaluated by PDH histogram analysis [35] and RS analysis [36]. The PDH histogram analysis is implemented in the following way: In the image, pixels are gathered into various classes of two consecutive pixels [35]. In each class, their variation value is found. Each of these variation values falls in the range −255 to +255. The PDH histogram analysis is based on the pixels differences. Here, the PDH denotes a graph with difference value on X-axis and its frequency range on Y-axis. The PDH of the original image usually will be a steady graph. If the stego-image is not a smooth graph, then the data is detected easily. If the PDH graph of the stego-image is in continuous nature, then the data hiding technique is not identified.

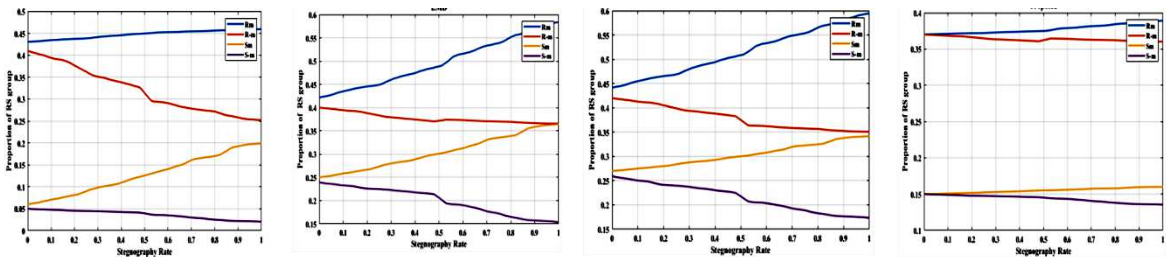


(a) (b) (c) (d)

Fig.12. PDH histogram differences between cover frames and theirs corresponding stego frames: (a) LSB, (b) Image+I, (c) RDHI and (d) RDHI and PVD using PETS09-S2L1 database video.

The PDH analysis of proposed RDHI+PVD and existing technique over 4 test videos is illustrated in Fig. 12. The solid line curve shows the PDH graph of the input video (frame 15), and the dotted line curve shows the PDH of the equivalent output video (frame 15). For the proposed frame Fig 12 (d), it is very much evident that there is no fluctuation in any of the stego-image. Hence, it is reliable to estimate that the proposed RDHI+PVD data hiding technique which is undetectable by PDH analysis for all the frames in the database.

The RS analysis [36] is accomplished using the four parameters R_m , R_{-m} , S_m and S_{-m} . If the condition $(R_m \approx R_{-m}) > (S_m \approx S_{-m})$ is satisfied, then RS analysis cannot detect the data and if the condition $(-R_{-m} \approx -S_{-m}) > (R_m \approx -S_m)$ is solved, then it proves that the RS analysis is able to identify the data.



(a) (b) (c) (d)

Fig.13. RS analysis for cover frames and the stego frames produced by (a) LSB embedding technique, (b) Image+I, (c) RDHI and (d) RDHI and PVD method using PETS09-S2L1 database video.

The RS analysis on input (frame 15) and output (frame 15) videos using PETS09-S2L1 database is shown in Fig. 13. In addition, it is clearly seen that the existing methods with increase in capacity the condition $(-R_{-m} \approx -S_{-m}) > (R_m \approx -S_m)$ is accurate. The R_m , R_{-m} as well as, S_m and S_{-m} shows much variation and generate a clear signature of the embedded data. From Fig. 13(d), it can be observed that for the proposed RDHI and PVD technique, the condition $(R_m \approx R_{-m}) > (S_m \approx -S_m)$ satisfies for both frames and all frames in video. From the result of these two frames, it can be predicted that the proposed technique resists the RS analysis.

4. Conclusion

This paper proposed pattern-based color video data hiding technique for the prediction of motion regions in videos. The proposed RDHI and PVD data hiding technique based on weights helps to sustain the video quality. Here, paring the moving pixels using Euclidean distance, and the data is embedded inside the moving objects are done by pattern. The proposed system has been evaluated in the PETS 2009 database. When compared with the state-of-the-art methods, this framework has revealed that the performance of the proposed system could increase the range of data embedding by utilizing three layer of data hiding and improve the video quality with fewer distortions. Furthermore, the PDH analysis and RS analysis have been implemented to examine the attack resistance of the proposed technique. It is clearly noticed that the PDH graph of the stego images do not appear in zigzag manner. Therefore, PDH analysis fails to detect the hidden data. Moreover, it is proved that the RS analysis fails to find the hidden data. In future, the security can be further improved by applying encryption techniques to the secret message.

References

[1] X. Ma, Z. Pan, S. Hu, and L. Wang, High fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism, J. Vis. Communication Image Represent, vol. 28, pp. 71_82, Apr. 2015.

[2] H.-T. Wu, S. Tang, J. Huang, and Y.-Q. Shi, A novel reversible data hiding method with image contrast enhancement, Signal Process., Image Communication., vol. 62, pp. 64_73, Mar. 2018.

[3] Mritha Ramalingam, Nor Ashidi Mat Isa, A data-hiding technique using scene-change detection for video Steganography, Computers, and Electrical Engineering, no.54, 423–434, 2016.

[4] Ekta W, Payal J, Navdeep. An analysis of LSB & DCT based steganography. Global Journal Computer Science Technology,10(1), 2010.

[5] C.-K. Chan and L. M. Cheng, Hiding data in images by simple LSB substitution, Pattern Recognition., vol. 37, no. 3, pp. 469_474, Mar. 2004.

[6] R.-Z. Wang, C.-F. Lin, and J.-C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognit., vol. 34, no. 3, pp. 671_683, Mar. 2001.

[7] S. Dumitrescu, X. Wu, and Z. Wang, Detection of LSB Steganography via sample pair analysis, IEEE Trans. Signal Process., vol. 51, no. 7, pp. 1995_2007, Jul. 2003.

[8] J. Fridrich, M. Goljan, and D. Rui, Detecting LSB steganography in color, and gray-scale images, IEEE Multimedia, vol. 8, no. 4, pp. 22_28, Oct./Dec. 2004.

[9] Ni Z, Shi YQ, Ansari N, et al. Reversible data hiding. IEEE Trans Circuits System Video Technology,16 (3):354–362, , 2006.

[10] X. Zhang and S.Wang, Efficient steganographic embedding by exploiting modification direction, IEEE Commun. Lett., vol. 10, no. 11, pp. 781_783, Nov. 2006.

[11] Chang CC, Chou YC, Kieu TD, Information hiding in dual images with reversibility, Proceedings of Third International Conference on Multimedia and Ubiquitous Engineering, Qingdao, China, 145–152, 2009.

[12] X. Zhang, W. Zhang, and S. Wang, Efficient double-layered stegano-graphic embedding, Electronic Letters vol. 43, no. 8, pp. 482_483,Apr. 2007.

[13] X. Zhang, W. Zhang, and S. Wang, A double layered `plus-minus one' data embedding scheme, IEEE Signal Process. Letters vol. 14, no. 11, pp. 848_851, Nov. 2007.

[14] C.-C. Wang, W.-C. Kuo, Y.-C. Huang, and L.-C. Wuu, A high capacity data hiding scheme based on re-adjusted GEMD, Multimedia Tools Appl., vol. 8, pp. 1_15, 2017.

[15] H.C. Wu, N.I. Wu, C.S. Tsai, M.S. Hwang, Image steganographic scheme based on pixel-value differencing and LSB replacement methods, IEEE Proceedings Vision, Image and Signal Processing, 152 (5), 611–615, 2005.

[16] X. Liao, Q.Y. Wen, J. Zhang, A steganographic method for digital images with four-pixel differencing and modified LSB substitution, Journal of Visual Communication and Image Representation, 22 (1), 1–8, 2011.

[17] G. Swain, A steganographic method combining LSB substitution and PVD in a block, Procedia Computer Science, 39–44, 2016.

[18] G. Swain, High capacity image steganography using modified LSB substitution and PVD against pixel difference histogram analysis, Security and Communication Networks, 1–14, 2018.

[19] S.Y. Shen, L.H. Huang, A data hiding scheme using pixel value differencing and improving exploiting modification directions, Computers & Security, 48, 131–141, 2015.

[20] Gandharba Swain, Two new steganography techniques based on quotient value differencing with addition-subtraction logic and PVD with modulus function, Optik - International Journal for Light and Electron Optics 180, 807–823, 2019.

[21] MrithaRamalingam, Nor Ashidi MatIsa, A data-hiding technique using scene-change detection for video Steganography, Computers & Electrical Engineering, Volume 54, Pages 423-434, August 2016.

[22] Ramadhan J. Mstafa, Khaled M. Elleithy and Eman Abdelfattah, A Robust and Secure Video SteganographyMethod in DWT-DCT Domains Based on Multiple Object Tracking and ECC, IEEE access, 2017.

[23] Jung KH (2018) A survey of interpolation-based reversible data hiding methods. Multimedia Tools and Applications 77(7):7795–7810

[24] Jung KH, Yoo KY (2009) Data hiding using image interpolation. Computer standards and interfaces 31: 465–470.

[25] Aruna Malik, Geeta Sikka and Harsh K Verma, A Reversible Data Hiding Scheme for Interpolated Images Based on Pixel Intensity Range, Multimedia Tools and Applications, 2020.
[26] Ekta W, Payal J, Navdeep, An analysis of LSB & DCT based Steganography,Global Journal Computer Science Technology, 10(1), 2010.
[27] A. Yilmaz, O. Javed, and M. Shah, Object tracking: A survey, ACM Computer. Survey, vol. 38, no. 4, pp. 1_45, 2006.
[28] Ramadhan J. Mstafa, Khaled M. Elleithy and Eman Abdelfattah, A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC, volume 5, IEEE, 2017.
[29] Vasileios Karavasilis, Christophoros Nikou, Aristidis Likas, Visual tracking using the Earth Mover's Distance between Gaussian mixtures and Kalman filtering, Image and Vision Computing, vol. 29, 295–305, 2011.
[30] Weiqi Luo, Fangjun Huang and Jiwu Huang, Edge Adaptive Image Steganography Based on LSB Matching Revisited, IEEE transactions on information forensics and security, vol. 5, no. 2, June 2010.
[31] J. Ferryman and A. Shahrokni, ``PETS2009: Dataset and challenge,'' in Proc. 20th IEEE Int. Workshop Perform. Eval. Tracking Surveill., pp. 1_6, Dec. 2009, doi: 10.1109/PETS-WINTER.2009.5399556.
[32] A. Ichigaya, M. Kurozumi , N. Hara, Y. Nishida, E. Nakasu, A method of estimating coding PSNR using quantized DCT coefficients, IEEE Transactions on Circuits and Systems for Video Technology , Volume: 16 , Issue: 2 , Feb. 2006.
[33] Girod, B, What's wrong with mean-squared error? In A. B. Watson (Ed.), Digital images and human vision, Cambridge: MIT Press, pp. 207–220, 1993.
[34] Y.K. Lee and L.H. Chen, High capacity image steganographic model, Volume 147, Issue 3, p. 288 – 294, June 2000.
[35] Swain, G. Very High Capacity Image Steganography Technique Using Quotient Value Differencing and LSB Substitution. Arab J Sci Eng 44**,** 2995–3004, 2019.
[36] Nabanita Mukherjee, Goutam Paul, Sanjoy Kumar Saha & Debanjan Burman, A PVD based high capacity steganography algorithm with embedding in non-sequential position, Multimedia Tools Application, 2020.
[37] Chi-KwongChan, L.M.Cheng, Hiding data in images by simple LSB substitution, Pattern Recognition, Volume 37, Issue 3, Pages 469-474, March 2004.
[38] Mohammad AA, Al-Haj A, Farfoura M, An improved capacity data hiding technique based on image interpolation, Multimedia Tools and Applications, 2018.