



RI-CDVS: Robust and Imperceptible Compressed Domain Video Steganography Using H.265 Codec

Shamal Salunkhe¹ · Surendra Bhosale²

Received: 10 September 2022 / Accepted: 10 January 2023 / Published online: 26 April 2023
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2023

Abstract

The development of steganography methods has raised growing worries about steganography abuse. As the significant demand for digital video processing is on the rise from last decade, data security becomes a crucial issue. Motion vector manipulation (MVM)-based video steganography has caught attention since it can result in indirect and arbitrary alterations in video data. The moderate payload capacity and complexity are issues faced by MV-based methods. A hybrid motion estimation and transform coefficients strategy applied on video steganography using the H.265 compression method is proposed. The robust imperceptible compressed domain video steganography (RI-CDVS) model is presented to increase imperceptibility with improved security. The two phases of the RI-CDVS model are embedding and extraction. The embedding stage generates the compressed stego video from the inputs of compressed cover video and secret image. Using dynamic threshold from the cover video, the motion estimation technique is used to select the group of key frames. The key frames are chosen to hide the secret image without sacrificing quality and lower error rate. The Discrete Cosine Transform (DCT) is used to transform keyframes into the frequency domain. The Least Significant Bit (LSB) of the integer coefficients of the DCT components is used to embed the secret information. The H.265 codec is used to create the compressed stego video. At extraction phase reverse operations are performed to get secret image. The experiments are conducted using a publicly accessible video collection and compared the results of RICDVS with the techniques at the cutting edge of video steganography.

Keywords Compression · Embedding · Extraction · H.265 codec · Motion estimation · Video steganography

Introduction

Digital video has emerged as one of the most prominent media because of highly interactive Internet of Things (IoT)-based multimedia applications [1]. Effectiveness of steganography depends on embedding efficiency, concealment capability, imperceptibility, and robustness [2]. Based on the embedding domain, video steganography is parted into two

categories: compressed-domain and uncompressed-domain video steganography [3]. The video stream is composed of a series of still images that are displayed consecutively and are timed out in the same manner. The same techniques that are used for steganography of images are also applied to steganography of videos. Once the hiding capacity has been increased, a cover file of a smaller size can be utilised for the purpose of concealing the secret message. Thus, a produced stego-file is more manageable in terms of both its size and ability to be communicated. The video files are much complex in comparison to the image file, thus videos offer a higher level of protection against the attacker. A high degree of key frame redundancy is employed to achieve improved levels of security. In uncompressed-domain steganographic techniques, data embedding comes before compression so it is susceptible to the loss of concealed data caused by video compression. In compressed-domain video, steganographic data are embedded as per the syntax sections of the compressed video. The frequently used syntax elements are motion vectors, intra-prediction modes, inter-prediction

This article is part of the topical collection “Enabling Innovative Computational Intelligence Technologies for IOT” guest edited by Omer Rana, Rajiv Misra, Alexander Pfeiffer, Luigi Troiano and Nishtha Kesswani.

✉ Shamal Salunkhe
shamal.salunkhe@rait.ac.in
Surendra Bhosale
sjbhosale@ee.vjti.ac.in

¹ Instrumentation Engineering, R.A.I.T., Navi Mumbai, India

² Electrical Engineering, V.J.T.I., Mumbai, India

modes, quantization parameters (QPs), and quantized transform coefficients (QTCs). Prior to compression, message bits are sent using raw spatial domain pixels that have been modified using methods common to image steganography [2]. By modifying the states or values of the aforementioned video elements, embedding is carried out during compression [4] using intra-prediction modes, inter-partition modes (PMs), motion vectors (MVs), DCT coefficients, and quantization parameters (QPs). Encrypted data look like random noise along with background noise; a statistical analysis program will not be able to detect it. The most commonly used video distortion level metrics is the peak signal-to-noise ratio (PSNR). The statistical characteristics of video entities produced by a conventional video encoder are influenced because embedding in these domains is often linked with the compression process. After compression encoded bit-stream [5] is a popular embedding domain in which the syntax parts of entropy coding are altered to signify the concealed message bits. It leaves embedding artifacts or distortions in the bit-stream domain, which are used to perform steganalysis with the statistical properties of encoded coefficients.

Data embedding frequently uses compressed domain parameters are MV and prediction noise coefficients [6]. The MV information has been used for covert concealment because it is conveyed without loss during video compression. A crucial step in MP-based steganography is choosing the right MV to embed. Searching through potential candidate MVs is an important process since random changes in MV can cause large prediction mistakes. The greater prediction error is carried by MV and a large MV is typically the best choice for embedding. Some methods alter the phase angle of MVs to embed data. The steganographic technique which did not properly examine video motion features, recompression attack that makes changes in the prediction error block brought on by embedding, bit increase rate, and other factors should considered in MV-based techniques.

The major contribution of this research is to design and develop the RICDVS for video steganography. It is a unique technique to efficiently predict video motions in the compressed domain. Uncompressed video steganography techniques are more vulnerable to noise, compression, and decryption also pre processing is required to maintain security level. Aimed at these shortcomings RICDVS is proposed. Data compression increases payload capacity of cover video. The most recent video coding standard, High-Efficiency Video Coding (HEVC), is also known as H.265. It offers the maximum compression efficiency as compared to its forerunners, H.264 [5]. The reverses operations are carried out during the extraction phase to obtain the original cover video and secret image with the best quality.

This paper is organized as below. Studies of comparable works are presented in the “[Motivation](#)”. The design of the suggested methodology is presented in the “[The Proposed](#)

[RI-CDVS Method](#)”. The simulation results and discussions are presented in the “[Results and Discussion](#)” section. Finally, the conclusion is presented “[Conclusion](#)” section.

Motivation

The traditional raw steganography techniques pose many limitations in terms of embedding capacity, robustness and imperceptibility against several attacks. These observed problems in the existing video steganography are stimulated for the development of proposed RICDVS. The novel video steganography approach is with maximum payload capacity and security with H.265 codec.

Literature Review

The aim of this research is to review recent studies on video steganography, analyze their limitations, and seek solutions to overcome these challenges.

A novel approach for video stenography based on modified VMs had been proposed in [7]. Each local optimal MVs in the search area analyzed to find all local optimum MVs. Then modified MVs were selected for better video compression efficiency. The DCT and Discrete Transform Wavelet (DWT) were used. H.265 Codec was used to develop robust video steganography. The secret image was pre-processed using the Bose and Hamming, Chaudhuri, and Hocquenghem codes. Motion-based multiple tracking technique was used to isolate regions of interest for secret information encoding. Without considering noise, the maximal PSNR values were obtained for DCT 48.67 and DWT 49.01. The MV based with Homogeneous Block Selection (MV-HBS) approach had been proposed in [8]. The MVs of the homogeneous regions of the reference frames had been chosen for embedding the secret data. To improve imperceptibility, an efficient search window and polar orientation-based embedding technique were applied. The PSNR 40.9 dB with SSIM 0.988 are obtained. The video steganography (VS) using the optimization of pixel prediction was proposed in [9] called WEWO (Water wave–Earth worm Optimization) with Deep Recurrent Neural Network (DRNN). It had been designed to hide secret data in optimum pixels of key frames. It was a hybrid evolutionary algorithm. The proposed methodology was performed better PSNR 43.87dB with impulse noise and Correlation coefficient 0.973. A video steganography approach was developed as Oppositional Grey Wolf Optimization (OGWO) with DCT, and DWT techniques [10]. Scene alterations were used to identify the keyframes that were used to conceal the secret data. The DCT was used to detect scene changes. OGWO had used to choose the best place to hide secret data once the keyframes were detected. The pre-processing of secrete image is performed to strengthen

security and eliminate video distortions. The findings of the experiments obtained a PSNR value of 75.141 dB. The Non-dynamic Region (NR) was extracted from the input video. The Discrete Sine Transform (DST) applied NR of frames followed by secret information embedding with the LSB technique. Then stego frames compressed using the H.264 codec and transmitted. The DST-Secret Bit Positions of NR for Message (SBPNRM) were proposed [11]. This non-dynamic region is converted from the spatial domain to the frequency domain by the DST. To protect the confidentiality of the data, the LSBs of the integer portion of the DST components are utilised. The game theory optimization mechanism was designed for efficient video steganography. To get the best solution, Iterative Elimination of Strictly Dominant Strategies (IESDS) was used [12]. The term pay-off was used for quantitative measure of Concealing Capacity (CC) and Peak Signal-to-Noise Ratio (PSNR). The best obtained values of CC and PSNR were 56.888 and 0.9989 dB, respectively. Transform Block Decision approach was developed for Video Steganography (TBDVS) [13]. Embedding error and modified transform block decision were analyzed to hide secret information and updated corresponding residuals. Better visual quality and huge embedding capacity were achieved by measurement of distortion, number of bits representing the residual samples in the decoder, and values of secret message.

Challenges

The video steganography in the compressed video domain (using H.264) is widely preferred in the above studies. The techniques like MVs, DCT, DWT, DST, FFT, etc. were used for the effective video steganography. Though MV-based techniques are promising, MV-based solutions are

underused in existing works. The MV-based challenges are listed below,

1. Selecting MVs that are as unnoticeable as feasible after alteration,
2. Devising a modification procedure that results in the fewest changes in the statistics of the final video,
3. Moderate payload capacity with moderate complexity, and
4. Vulnerable to the security threats [14].

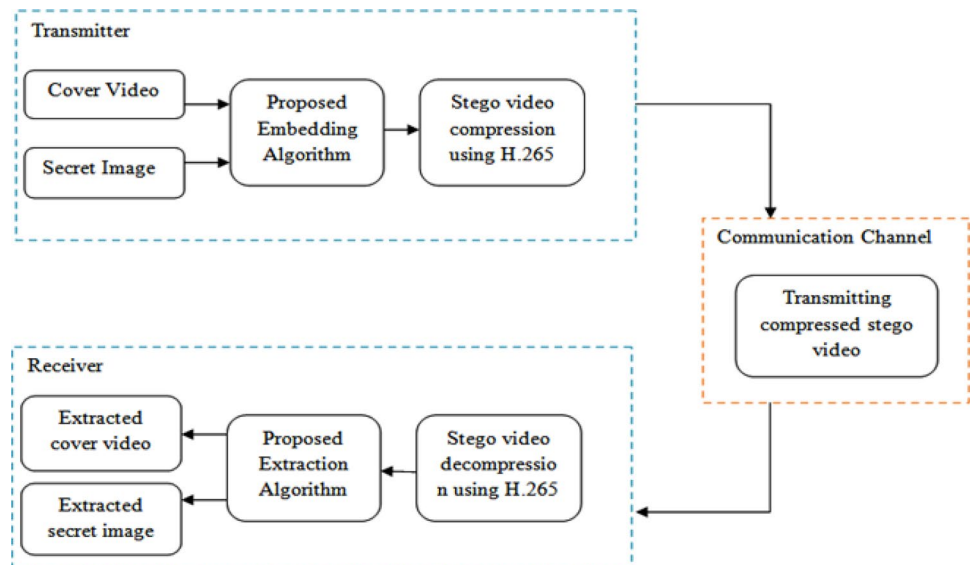
The key requirement of the video steganography method is to achieve the trade-off among perceptible invisibility, robustness, and payload and embedding capacity. The existing solutions, therefore, are required to modify for highly secure video steganography to achieve imperceptibility, robustness, and higher capacity.

The Proposed RI-CDVS Method

This section presents the complete methodology and design of the proposed RI-CDVS. Figure 1, demonstrates the architecture of the RI-CDVS. It consists of three blocks transmitter, communication channel, and receiver blocks.

The transmitter block has the cover Video and the secret image as an input. The proposed embedding algorithm is applied to hide the secret image into the cover video with minimum distortion and maximum embedding capacity. The stego video has then compressed using the H.265 technique, it is the outcome of the transmitter block. The compressed stego video has transmitted towards the receiver via a wireless channel. At the receiver block, the received stego video has decompressed using the H.265. The decompressed stego

Fig. 1 The architecture of the RI-CDVS



video and secret image are separated using proposed extraction algorithm.

Transmitter Block

The proposed embedding process is depicted in Fig. 2. The developed Dynamic Key frame Extraction (DKE) algorithm is used to select appropriate cover video frames from a sequence of video frames. The frames with non-motion regions with few motion regions are search using the dynamic thresholding technique. The selected key frames for video named Bowling are 3, 21, 26, 35, 43, 50, 59, 68, 83, 131, 142, 150, 159, 204, 219, 231, 255, 272 and 293.

The dynamic threshold is obtained for each input video using the histogram technique. It estimates the maximum number of keyframes compared to other recent keyframes or Region of Interest (ROI) extraction techniques. The cover video sequence C consists of n number of frames $F = f^1, f^2, \dots, f^m$ and input RGB secret image I . Key frames F are extracted from the input cover video C . The dynamic threshold (DT) using the histogram difference approach is used for ROI selection. The image histogram technique is used to estimate motion vectors in the form of distribution of the input frame [15]. The applied histogram technique for each frame is reliable and lightweight.

The histogram features are extracted for each frame $F(i) \in C$ and difference of histogram among each consecutive frames $F(i - 1) \& F(i) \in C$ compared with dynamically computed threshold value DT. If the frames pair $F(i - 1) \& F(i)$ has lower value for histogram differences compared to DT. Then non-dynamic regions of selected keyframes are used to embed the secrete information. Thus, a set of keyframes K is extracted for further processing. The DKE algorithm extracts the distinct keyframes with keeping the video sequence temporal ordering. The process of computing the histogram and its differences for two consecutive RGB frames $F(i - 1) \& F(i)$ are given below,

$$t^1 = \frac{\sum_{j=1}^3 (imhist(F(i - 1), j))}{3} \tag{1}$$

$$t^2 = \frac{\sum_{j=1}^3 (imhist(F(i), j))}{3}, \tag{2}$$

where j belongs to RGB channels data. Mean histogram is computed for every RGB frame. The difference of histogram for $F(i - 1) \& F(i)$ is computed by

$$t(i) = |t^1 - t^2| . \tag{3}$$

The difference value t is compared with dynamically computed threshold value DT. If value of t is lower than DT, then $F(i)$ is detected as non-dynamic frame and stored into the vector K .

Using 2D-DCT, extracted keyframes $k \in K$ are transformed from the spatial domain to the frequency domain. The intensity values of RGB component of k key frame is transformed into integer and fractional values using DCT. A DCT of an input frame is a sum of cosine with different magnitudes and frequencies. Visually important information about image is represented with few DCT coefficients. Thus, DCT is frequently employed in video processing applications. The 2D-DCT of DCT coefficient $d_{a,b}$ of stego key frame k_{ij} key frame of size $m \times n$ is computed as follows.

$$d_{a,b} = \alpha_a \alpha_b \sum_{i=1}^m \sum_{j=1}^n k_{i,j} \cos \frac{\pi(2i + 1)a}{2m} \cos \frac{\pi(2j + 1)b}{2n}, \tag{4}$$

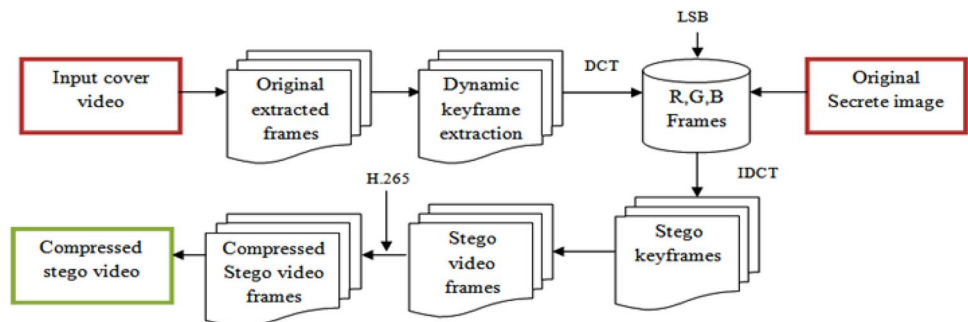
where $a = 1, 2, \dots, m$ and $b = 1, 2, \dots, n$. The coefficients α_a and α_b are computed as

$$\alpha_a = \begin{cases} \frac{1}{\sqrt{m}}, a = 0 \\ \frac{2}{\sqrt{m}}, 2 \leq a \leq m \end{cases} \tag{5}$$

$$\alpha_b = \begin{cases} \frac{1}{\sqrt{n}}, a = 0 \\ \frac{2}{\sqrt{n}}, 2 \leq b \leq n. \end{cases} \tag{6}$$

The k_{ij} represents intensity value of pixel at location (i, j) in spatial domain for the current key frame $k \in K$ and $d_{a,b}$

Fig. 2 The proposed embedding process



represents the same pixel value in frequency domain by applying DCT.

The DEK algorithm identifies keyframes and 2D-DCT is applied on key frames to convert RGB components of frame into a frequency domain. The LSB approach is used to conceal information in each DCT coefficient. While using inverse DCT, the integer part of the DCT value had taken into account to hide a secret image with a lower possibility of changing the original intensity value of a pixel. The LSB of the binary form of the integer part is retrieved. The IDCT is a mathematical formula that converts pixel intensity values from the frequency domain to the spatial domain. The 2D-IDCT of each DCT coefficient $d_{a,b}$ of stego key frame $k_{i,j}$ of size $a \times b$ is computed as follows:

$$k_{i,j} = \sum_{a=1}^m \sum_{b=1}^n \alpha_a \alpha_b d_{i,j} \cos \frac{\pi(2a+1)i}{2m} \cos \frac{\pi(2b+1)j}{2n}, \quad (7)$$

where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. The coefficients α_a & α_b are computed using Eq.(5) and (6). The $d_{a,b}$ represents intensity value of pixel at location (a, b) in frequency domain for the current key frame $k \in K$ and $k_{a,b}$ represents the same pixel value in frequency domain after applying IDCT. The stego keyframes and other frames of the input cover video are combined according to their original temporal sequence to produce the stego video. The H.265 compression is applied on stego cover video to produce the compressed stego cover video.

H.265 (HEVC) is the most recent video coding standard and an upgrade of H.264 called Advanced Video Coding (AVC). It gives a better picture quality and compression efficiencies to make massive data files more manageable and minimize the storage burden. H.265 can reduce bit rate requirements and associated storage needs by around 30% with no apparent reduction in video quality. Instead of encoding every pixel from every frame, H.265 compression identifies static areas. This compression technique has also improved capabilities such as motion compensation, spatial prediction, and sample adaptive offset (SAO) picture filtering. The motion compensation is a process utilized for predicting the frame from a video provided the previous and future frames based on motion of camera or objects in the video. The programming language Python is used to run a simulation of compress domain video steganography. Live streaming with H.265 offers a higher quality while using less bandwidth than other formats. H.265 is able to compress the data in live video streams in a more effective manner. The video bandwidth of a video stream can be calculated by multiplying the resolution of the image that was captured by the total number of frames that have been in the video. Constant Bit Rate (CBR) is the procedure that is used by all video compression standards. The greater the quality of the video stream that is being transmitted, the higher the Bit

Rate that is responsible for it. An internet speed of at least 1.5 Mbps is required to stream videos of a 4K resolution quality using the H.265 codec.

Intra-frame and inter-frame prediction, motion estimation, 2-D discrete transform, and encoding steps are used in H.265/HEVC video coding. The algorithm takes raw frames and separates them into groups (GOP). Each GOP has a set of number of frames. Intra-picture prediction codes the initial frame of a GOP. The remaining GOP frames are coded using inter-picture prediction in one of two modes,

1. Prediction model (P-frames) and
2. Bi-directional prediction (B-frames).

Each GOP starts with an I-frame and ends with P- or B-frames. Motion vectors associated with each block of the current image are found using inter-picture (P and B frames) encoding. Motion compensation assumes that many blocks in a video sequence will move relative to the I-frame, so rather than recording all the blocks, it is sufficient to encode which blocks moved how much and in which direction.

The residual signal macro block is the difference between the original and predicted blocks, and is mathematically converted using spatial transform. The transform coefficients are then scaled, quantized, and entropy coded. The result is transferred in HEVC format.

The key steps of the H.265 coding standard are briefly described below which shows the improvement over the previous compression versions.

1. Hybrid Prediction: The residual macro block is predicted using both intra- and inter-frame prediction methods. The predicted residual signals are further arranged in a quad-tree manner. As the HEVC Coding Tree Units (CTU) consist of chroma and luma coding tree blocks (CTB), their size can be extended to 64×64 .
2. Motion Estimation: Motion estimation finds matching pixel blocks in inter-frame coding. Although there is no motion because the blocks are matched within a single frame, the goal to exploit data redundancy. Inter-frame prediction predicts the motion of pixel blocks to discover temporal redundancy between two consecutive frames. By introducing much narrower angles of supported orientations, H.265/HEVC significantly enhanced intra-prediction. There are thirty three non-uniform angular prediction modes in HEVC. Near-horizontal and near-vertical angles are finer, while diagonal angles are coarser. This configuration allows for improved statistical matching of pixel blocks across frames.
3. Transform and quantization: The H.265/HEVC standard specifies two types of transforms: a core transform (on inter-prediction mode) and an alternate transform (intra-prediction mode). The DCT is the core transform,

which is applied to 4×4 , 8×8 , 16×16 , and 32×32 transform blocks (TB) using Eq. (4). The alternative transform is derived from DST, is employed only for 4×4 luma residual TBs in the intra-picture prediction mode using Eq. (8). The DCT and DST coefficients are quantized by adjusting the quantization parameter.

4. Encoding: H.265 supports only context-adaptive binary arithmetic coding (CABAC) compared to previous standards. For each scenario, CABAC uses distinct probability models to encode entropy losslessly. Because local data are often well-correlated, this allows for better modeling of distribution.

The 2D-DST $d_{a,b}$ of each inter-prediction block $b_{i,j}$ of size $m \times n$ is given by

$$dst_{a,b} = \frac{4\alpha_a\alpha_b}{mn} \sum_{i=1}^m \sum_{j=1}^n b_{i,j} \sin \frac{\pi(2i+1)a}{2m} \sin \frac{\pi(2j+1)b}{2n}, \quad (8)$$

where $a = 1, 2, \dots, m$ and $b = 1, 2, \dots, n$ represents height and width of the block b. The coefficients are computed using Eqs. (5) and (6). The detailed procedure of the video embedding and compression is represented in figure 2. The first step is to divide the input RGB secrete image I into the number of portions according to the discovered keyframes. To discover the portion of each channel of a secrete image, the following Eq. (9) is used:

$$I_i^p = \frac{3 \times I_i}{count^1}, \quad (9)$$

where I_i represents the i^{th} channel of input secrete image I and $count^1$ represents the total number of keyframes discovered into the input cover video C . The variable p represents the portion of the cover image, $p = 1, 2, \dots, count^1$.

The objective is to hide each portion of each channel of input secrete image into the LSB part of the DCT coefficient of each channel of each key frame such that it leads to

higher embedding capacity and less video distortion. The *embed* function takes the LSB of the DCT component and the current portion of a secrete image of each channel. The embedding function works with 2D DST with cosine component. After embedding the portion of each channel of the secrete image into each key frame corresponding channel, the concatenate function *cat* is used to generate the stego cover frames.

The $S(1, j)$, $S(2, j)$ and $S(3, j)$ represent the stego RGB channels of a j^{th} cover video frame. The *cat* function generates the *temp* RGB frame by, $temp = \sum_{i=1}^3 S(i, j)$, where $j = 1$ to $count^1$.

The stego cover *SV* video is generated by replacing each key frame in the original video sequence with stego keyframes. On the stego cover video, we applied the H.265 compression technique to generate the compressed stego cover video (CSV). The steps of the H.265 codec consist of hybrid prediction, motion estimation, transformation-quantization, and encoding. The outcome of the transmitter block is CSV is then transmitted towards the receiver block.

Receiver Block

The extracting process of the proposed RICDVS extracts the compressed secret image from the compressed stego video. It is an inverse procedure of embedding; the processing steps are shown in Fig. 3.

As shown Fig. 3, the input for the receiver block is the H.265 CSV stream. The CSV first decompressed using the inverse H.265 operations such as hybrid prediction of inter-prediction and intra-prediction blocks, inverse transformations of inter-prediction TBs and intra-prediction TBs using 2D-IDCT using Eq. (7) and 2D-IDST using Eq. (10), inverse quantization is applied to get the decoded residual signals, and finally applied inverse CABAC to get the decompressed stego frames. The 2D-IDST $b_{i,j}$ of each DST inter-prediction block of size $m \times n$ is given by

Fig. 3 The block diagram of extraction process

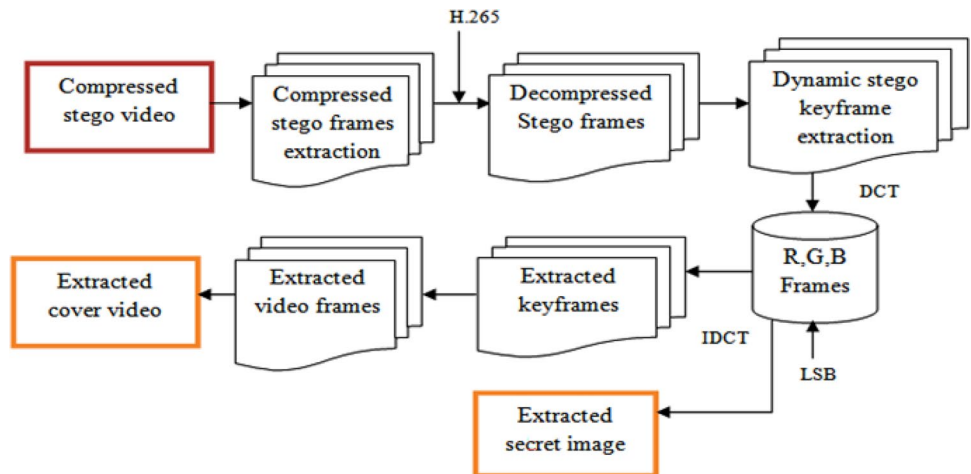


Table 1 Comparative analysis for APSNR

Pairs	WEWO	IESDS	MV-HBS	TBDVS	RI-CDVS
1	45.7240	46.0398	47.9906	49.8276	52.8704
2	44.0481	45.7356	47.2153	48.5139	51.5586
3	44.9345	46.5487	47.9248	49.6731	52.6872
4	45.1193	46.9847	48.7119	50.7539	53.7950
5	44.5712	45.1738	46.8200	47.612	50.7154
6	44.1069	44.9557	46.7694	47.4592	50.4748
7	43.5871	44.7395	46.2279	47.3365	50.4209
8	44.8607	45.9600	47.7913	49.6027	52.6830
9	44.3924	46.5163	48.0586	49.7920	52.1901
10	44.5307	45.7879	47.6681	49.9642	52.9985

Table 2 Comparative analysis co-relation coefficient

Pairs	WEWO	IESDS	MV-HBS	TBDVS	RI-CDVS
1	0.9704	0.9775	0.9801	0.9839	0.9897
2	0.9700	0.9722	0.9846	0.9806	0.9854
3	0.9622	0.9651	0.9829	0.9857	0.9888
4	0.9840	0.9789	0.9883	0.9896	0.9941
5	0.9768	0.9784	0.9810	0.9824	0.9832
6	0.9637	0.9769	0.9774	0.9800	0.9800
7	0.9715	0.9777	0.9821	0.9859	0.9867
8	0.9708	0.9734	0.9805	0.9836	0.9845
9	0.9643	0.9685	0.9772	0.9800	0.9807
10	0.9810	0.9825	0.9900	0.9928	0.9936

$$b_{ij} = \sum_{a=1}^m \sum_{b=1}^n \alpha_a \alpha_b d_{ij} dst \sin \frac{\pi(2i+1)(a+1)}{2m} \sin \frac{\pi(2j+1)(b+1)}{2n}, \tag{10}$$

where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$ represents height and width of the block dst . After decompression of CSV video using H.265, we applied DKE method to get the stego keyframes SK . Then, we have initiated the procedure of secrete bits extraction from the each channel of each key frame. After decompression and stego keyframes extraction, the numbers of secrete image bits are discovered T embedded into each channel of keyframes. The secrete image portion embedded in each channel of each frame is computed by,

$$T_i^p = \frac{3 \times T}{count^1}, \tag{11}$$

where $count^1$ represents the total number of stego keyframes discovered. The variable p represents the portion of the cover image such that $p = 1, 2, \dots, count^1$ for i_{th} channel.

The 2D DCT converts each channel of key frame k from the spatial domain to the frequency domain where the secrete image has been hidden. For each channel of each key frame, we have applied the 2D DCT k_{det}^i and computed LSB k_{lsb}^i . The secrete message is extracted from the integer component of LSB using the *extract* function. The process of extraction is exactly opposite to the *embed* function as mentioned in [11]. After secrete bits extraction $E_i(p)$ for each channel of each frame k , we applied the inverse 2D DCT to recover the original key frame. The reshape and concatenation operations are applied to build the extracted secrete image and extracted cover video. Because of these operations extraction time is increased and accuracy of extracted image affected slightly. The correlation coefficient is performance measure of encryption algorithm. There is a good correlation between original and secure image is measured.

Fig. 4 APSNR values for pairs

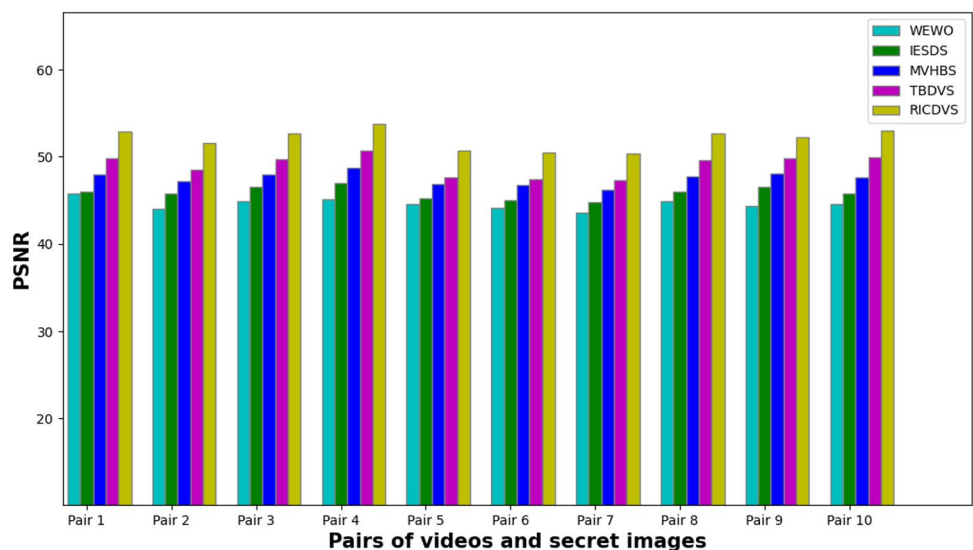
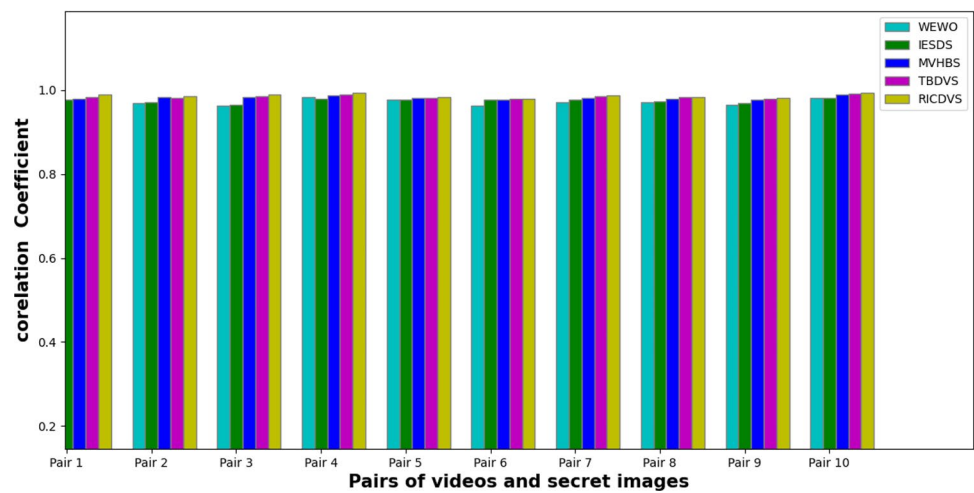


Fig. 5 Correlation coefficient values for pairs**Table 3** Comparative analysis for embedding time

Pairs	WEWO	IESDS	MV-HBS	TBDVS	RI-CDVS
1	21.78	21.34	18.99	19.45	17.81
2	20.10	21.23	16.41	18.78	15.91
3	19.44	20.39	16.56	17.56	15.54
4	18.89	19.45	16.56	16.99	15.48
5	24.27	23.49	18.41	19.78	18.01
6	22.68	23.23	20.31	20.31	18.57
7	21.02	21.79	18.77	22.95	18.11
8	22.36	21.45	18.59	23.51	18.16
9	19.92	21.72	18.92	22.46	18.17
10	19.34	19.59	17.45	21.81	15.61

Table 4 Comparative analysis for extraction time

Pairs	WEWO	IESDS	MV-HBS	TBDVS	RI-CDVS
1	12.57	14.22	12.66	13.31	11.38
2	15.97	14.15	11.63	12.85	11.02
3	14.29	13.59	11.04	13.12	10.32
4	13.18	12.96	12.37	13.01	11.17
5	15.32	15.66	12.94	14.52	11.73
6	14.67	15.48	13.54	14.93	12.12
7	14.02	14.51	12.51	13.78	11.69
8	16.31	14.38	12.37	14.23	11.71
9	15.46	14.43	12.61	14.39	11.78
10	14.00	13.72	11.63	13.48	10.92

Results and Discussion

This section describes the results and discussion of the proposed RICDVS with evaluation metrics, such as CC and

PSNR. The dataset and techniques at the cutting edge of video steganography, performance measures, comparative analysis and discussion on result are presented in this section.

Dataset

The videos of different resolutions are used as the cover videos and RGB images are used as secret images. The RGB images are collected from the MICC-F2000 dataset. For video sequences, two sources are used, Elecard video sequences and YUV video sequences. All cover videos are of different resolutions. The first three videos were collected from Elecard video datasets and the remaining seven videos were collected from the YUV dataset. For this experiment different films from these dataset were used, with resolutions ranging from 480×832 , 768×1024 , 720×1280 , and 800×1280 . The number of frames ranges from 150 to 350, with frame rates of 15, 30, 50 and 60 fps.

To analyze the efficiency of the proposed RICDVS model, we have compared the performances with recent similar methods such as MV-HBS [8], WEWO [9], IESDS [12], and TBDVS [13]. The brief functionality of these methods has already been discussed in “[Motivation](#)” section. These methods used for comparative analysis based on ROI and pixel prediction techniques in video steganography. The H.265 compression technique is used in TBDVS method.

Performance Measures

The comparative study among all these methods has been performed using the different performance metrics to analyze parameters such as PSNR and correlation coefficient.



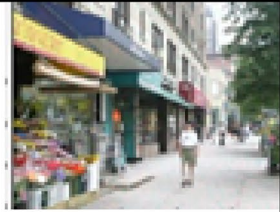




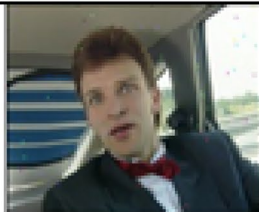

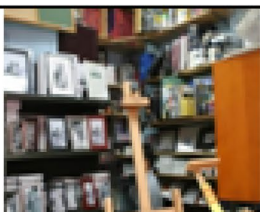


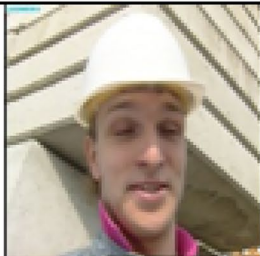
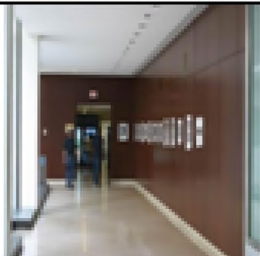

Name	Cover Video	Secrete Image	Stego Frame	Extracted Image
Akiyo				
Bus				
Carphone				
Football				
Foreman				

Fig. 6 Original, stego video frame with original and extracted secret image

Imperceptibility Measures

The APSNR computes the visual quality of original and stego video sequence. Higher APSNR and lower AMSE imply higher imperceptibility of the video steganography technique. The APSNR computed as average PSNR of original and stego video frames. Suppose that f_1 represents the original video frame and f_2 represents the stego video frame. MSE and PSNR represented by

$$MSE_{rr} = \frac{1}{m, n, l} \sum_{c=1}^{m, n, l} (f_1(m, n, l) - f_2(m, n, l))^2 \tag{12}$$

$$PSNR(r, s) = 10 \log_{10} \frac{\max(f_1)^2}{MSE}, \tag{13}$$

where m and n represent the height and width of each frame and l represents the three RGB channels. The AMES and APSNR are computed by averaging the MSE and PSNR

for all the original video frames and stego video frames. Table 1 shows the comparative analysis of APSNR value of RICDVS and other methods. Te RICDVS indicating substantial improvement in APSNR.

Figure 4 is graphical representation of the APSNR values. H.265 compression with proposed embedding algorithm RICDVS is showing better impressibility.

Robustness Measures

The robustness of the proposed video steganography method is analyzed using correlation coefficient (CC) parameter. The CC computed from the original secrete image I and extracted secrete image EI of size $m \times n$.

The correlation coefficients are a widely used as a performance evaluation measure in image processing. The correlation between two images is measured using correlation coefficients. Equation 14 represents the correlation coefficient as shown below,

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (14)$$

The statistical relation between two variables is given by correlation coefficient and it ranges between -1 and $+1$. Higher CC imply the higher robustness of the video steganography technique. Table 2 indicates comparison based on correlation coefficient. Figure 5 is graphical representation of average correlation coefficient of all methods. The proposed RICDVS is better in robustness behaviour.

Tables 3 and 4 are representing embedding time and extraction time in seconds required for every method. The proposed RICDVS has better computational efficiency. The RICDVS is finding out low dynamic ares and hiding secret image in it with less computational time.

Figure 6 is presenting few of the outcomes of experiments. Five videos and secret images and respective outputs are presented.

Conclusion

This paper proposed the novel approach for video steganography in the compressed domain using the H.265 codec. The proposed RICDVS is specially designed to overcome the limitations of existing video steganography solutions and improve the performance trade-off among the visual degradations, robustness, and embedding capacity. The complete methodology of the RICDVS mechanism is presented in this paper with the transmitter and receiver blocks. The novelty of the RICDVS is the algorithm of discovering the key frames for embedding the secrete information using the

lightweight motion estimation and dynamic threshold-based mechanism. This approach not only enhanced the embedding capacity but also improved the imperceptibility and robustness. Apart from this, the embedding and extraction algorithm had utilized the DCT-LST technique for hiding and extracting the secreted data. Based on video steganography investigation factors such as robustness, imperceptibility, and embedding capability, the empirical findings of the experiment demonstrate the efficacy of the suggested video steganography approach.

The performance investigation of the proposed model is compared to four recent similar methods in terms of APSNR, correlation coefficient, embedding time, and execution time. With improvement in the video steganography performance measures, the proposed model shows the computational efficiency as well. The future direction of the proposed model is to apply it to a large number of video samples. Artificial intelligence (AI) or swarm intelligence (SI) techniques to optimize video steganography will be another interesting research direction for this work.

Funding No funding sources.

Data availability The datasets used during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interest The authors declare that there is no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Weng X, Yongzhi L, Chi L, Yadong M. Convolutional video steganography with temporal residual modeling. *Comput Sci Multimedia Retrieval* 2018.
2. Mastafa R, Khaled. Video steganography techniques: Taxonomy, challenges and future directions. *IEEE LISAT* 2017;1–6.
3. Dasgupta K, Mondal JK, Dutta P. Optimized video steganography using genetic algorithm (ga). *Procedia Technol.* 2013;10:131–7.
4. Mstafa R, Elleithy K. Compressed and raw video steganography techniques: a comprehensive survey and analysis. *Multimed Tools Appl.* 2017;20:21749–86.
5. Chiu M, Siu Y. Computationally-scalable motion estimation algorithm for h.264/avc video coding. *IEEE Trans Consumer Electron.* 2010;56:895–903.
6. Zhang H, Cao Y, Zhao X. Motion vector-based video steganography with preserved local optimality. *Multimedia Tools Appl.* 2015;75:13503–19.

7. Mstafa R, Elleithy K, Abdelfattah E. A robust and secure video steganography method in dwt-dct domains based on multiple object tracking and ecc. *IEEE Access*. 2017;5:5354–65.
8. Rana S, Kamra R, Sur A. Motion vector based video steganography using homogeneous block selection. *Multimed Tools Appl*. 2020;79:5881–96.
9. Salunkhe S, Bhosale S. Nature inspired algorithm for pixel location optimization in video steganography using deep rnn. *Int J Eng Sci Technol*. 2021;3(2):146–54.
10. Suresh M, Shatheesh S. Optimized interesting region identification for video steganography using fractional grey wolf optimization along with multi-objective cost function. *Multimed Tools Appl*. 2021;80(9):13253–70.
11. Patel R, Lad K, Patel M, Desai M. A hybrid dst-sbpm approach for compressed video steganography. *Multimedia Syst*. 2021;27:417–28.
12. Suganthi K, Rajkumar S. A multi-image hiding technique in dilated video regions based on cooperative game-theoretic approach. *J King Saud Univ Comput Inf Sci*. 2021;34:5841–55.
13. Zhao H, Liu Y, Wang Y, Liu S, Feng C. A video steganography method based on transform block decision for h.265/hevc. *IEEE Access*. 2021;9:55506–21.
14. Zhang Z, Li Z, Liu J, Yan H, Yu L. Steganography algorithm based on modified emd-coded pu partition modes for hevc videos. *EURASIP J Image Video Proc* 2021;7.
15. Ghamsarian N, Schoeffmann K, Khademi M. Blind mv-based video steganalysis based on joint inter-frame and intra-frame statistics. *Multimed Tools Appl*. 2021;80:9137–59.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.