MP4 Video Steganography in Wavelet Domain

Hemalatha S, U. Dinesh Acharya, Shamathmika
Department of Computer Science and Engineering
Manipal Institute of Technology, Manipal University
Manipal, India
hema.shama@manipal.edu, dinesh.acharya@manipal.edu, shamathmika76@gmail.com

***Abstract*— Video steganography is a technique used for secret communication. In video steganography, the secret information is concealed in a video file. An MP4 video steganography method that hides audio and image data in wavelet domain is proposed in this paper. A video file consists of I-frame, P-frame, and B-frame. An I-frame, or intra frame, is decoded independently without referencing other frames. P-frames and B-frames are responsible for capturing the motion. A number of I-, P- and B- frames together is called Group of Pictures or GOP. A GOP starts with an I-frame whose length (number of frames between two I- frames, excluding the second I-frame) varies depending on the codec used. Since I-frames are not lost during compression or any kind of signal processing operations, they are selected for embedding. Appropriate pixels of the I-frames and the audio frames are used for embedding without affecting the perceptual features of the video. Redundant copies of the secret information are made to avoid frame loss. The image frames hide the secret images and the audio frames hide the secret audio signals. The secret image and audio signals are transformed using integer wavelet transform and the low frequency coefficients are hidden. The performance is analyzed and the effect of common attacks is discussed.**

Keywords—Video steganography; wavelet; I-frame; MP4; GOP

I. Introduction

Video transmission over the internet is common nowadays because of the availability of large bandwidth. Video files have large amount of redundancy, so huge quantity of secret data can be concealed and the distortion is unnoticeable here. The first frame in a video stream is an I-frame. A P-frame, or predictive inter frame, is coded using the previous I- and/or P- frame(s). It comprises the differences relative to previous I- or P- frames. A B-frame or bi-predictive inter frame makes references to I- or P-frames, preceding and succeeding it. Therefore, P-frames and B-frames are only concerned about capturing the motion. The Group of Pictures or GOP which is the group of I-, P- and B- frames starts with an I-frame. Its length (number of frames between two I-frames, excluding the second I frame) varies depending on the codec used. A typical GOP structure is IBBPBBPBBPBBPBB or IBBBPBBBPBBB. One GOP of length 9 is shown in Fig. 1. The I-frame predicts the first P-frame. The I- and the first P- frames predict the first two B-frames. The second P-frame is identified by the first P-frame and these two find the third and fourth B-frames [1].

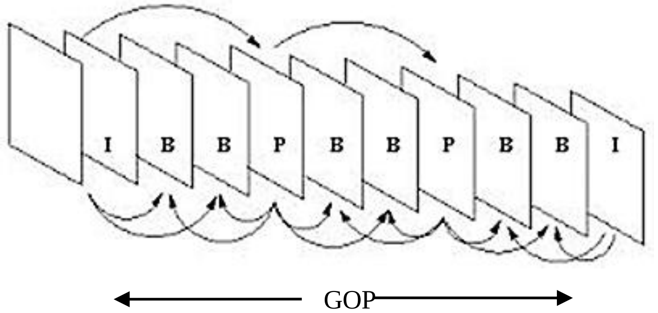


Fig. 1. Typical sequence of I-, P- and B- frames

The MP4 video files are highly compressed files and the header is only one byte. Only a few bits of this byte can be used for steganography. Therefore, the header is not a good candidate for information hiding. I-frames are not affected during compression and hence I-frames and audio frames can be used for embedding.

There are two categories of steganography techniques: spatial domain and transform domain techniques. Transform domain techniques are more robust than spatial domain techniques as they have high resistance against signal processing operations. Discrete Wavelet Transform (DWT) is the mostly used transform in the transform domain. Integer Wavelet Transform (IWT) is a DWT using which the original data can be reconstructed perfectly. In this paper, IWT is used for transforming the secret image and secret audio. When an image is transformed using IWT, it is divided into four components and when an audio signal is transformed using IWT, it is divided into two components. The low-frequency or approximation components contain the significant features of the signal [2], [3], [4]. In the following sections the literature survey, the proposed method and the results are explained. Section II provides the literature survey. In section III the proposed method, result analysis and the comparison of the proposed work with the related existing methods are discussed.

II. Literature Survey

Video steganography is an extension of the image steganography. There are several image steganography techniques applicable to videos as well. Since the video contents are dynamic, the chances of detecting the hidden data are less compared to the images.

N. Hideki et al. proposed two data hiding methods in the lossy compressed video: SPHIT-BPCS and MotionJPEG20000-BPCS, which are based on the video compression using wavelet and the Bit Plane Complexity Segmentation (BPCS) steganography [5]. SPHIT means 3-D set partitioning in hierarchical trees (SPIHT) video coding. In 3-D SPHIT video compression, the video is decomposed by 3-D Discrete Wavelet Transform (DWT) and the resulting wavelet coefficients are used for steganography through BPCS steganography. In Motion-JPEG2000 video coding, the I-frames are encoded using JPEG2000, which is based on 2-D wavelet compression. The BPCS steganography is applied on the resulting wavelet coefficients. The authors demonstrated that 3-D SPHIT-BPCS gives better capacity and video quality.

X. Changyong et al. proposed an efficient technique for steganography in the compressed video [6]. The control information necessary for extraction is stored in the I- frame and in P- and B- frames, the actual data is hidden repeatedly to resist the video processing operations. In MPEG compressed video, I-frames are DCT encoded, P- frames and B- frames are encoded to produce motion vector data stream. The P- and B- frames are considered to be having macro blocks. The P- frames have one motion vector per macroblock and the B- frames have two motion vectors per macroblock. Each GOP is divided into four segments and the data is embedded four times in these segments excluding the I-frame, by computing the embedding capacity of each macro block. This technique overcomes the frame loss or frame insertion but fails when the video file format is changed.

A. J. Mozo et al. used Flash Videos for steganography as its file structure is simple and its size is smaller than other video file formats [7]. The technique used here hides the information in the file structure such as in the tag field, at the end of the metadata etc. It has resistance to compression with 100% lossless extraction, good stego video quality, and perfect integrity of the hidden data. However, since the information is hidden only in the tag fields the embedding capacity in this method is low even though the cover video size is large and the information can be detected easily from the file structure

P. Feng et al. proposed a video steganography technique to hide the secret messages in the motion vectors of the cover media during H.264 compression process [8]. Linear block codes are used for reducing the changes in the motion vectors. A. A. Hussein proposed another video steganography technique on compressed video using motion vectors. Here, to encode and reconstruct the P-frames and B- frames in video compression, motion vectors are used [9]. The motion vectors are selected for embedding if their related macroblock prediction error is lesser than an initial threshold value Tmax. The threshold is selected in such a way that the hidden information is imperceptible. The secret information is hidden in the least significant bit of both the components of the selected motion vectors.

S. Zafar et al. proposed a high payload steganography technique for H.264/AVC video, which hides the data during the compression process [10]. The quantized transform coefficients, which are above a certain threshold, are considered for data embedding. This technique offers high payload, without targeting robustness.

In the steganography technique proposed by S. Po-Chyi et al. for H.264/AVC video, the video file is first decoded to get video and audio frames and then the information is hidden in the quantized coefficients, inter predicted motion vectors and intra predicted motion vectors [11]. The embedding is done during encoding the previously decoded frames. It achieves only 10% of the video file size as the capacity and it increases the stego files size. So, it is likely to the doubt about the hidden data. G. Sagar et al. proposed a MPEG-4 video steganography in DCT domain to enhance the capacity and perceptibility [12].

L. Yunxia et.al proposed a robust readable data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift [13]. The encoded data is embedded in the coefficients of the 4 × 4 luminance DCT blocks in I- frames and therefore the capacity is very low. S. Tamer proposed two steganography techniques for MPEG video [14]. In the first method, the quantization scale of a constant bitrate video is modulated to conceal the secret data bits. A capacity of one bit per macroblock is attained. The other method finds the association between the feature variables at macroblock level and the secret message bits using a second-order multivariate regression. The decoder uses the regression model to predict the values of the hidden message bits with very high prediction accuracy.

X. Dawen et al. attempted to conceal in the H.264/AVC video by encrypting the code-words of the intra-prediction modes, the motion vector differences, and the residual coefficients with the stream ciphers, and then embedded the secret data using the code-word substitution [15]. However, it did not give promising security. Y. Yuanzhi et al. defined embedding distortion for data hiding in motion vectors [16]. But, it increases the noise, which is not desirable. Hemalatha S et al. suggested a method to hide both image and audio data in uncompressed video, which is not robust against compression [17]

In the compressed video steganography techniques mentioned above, the payload capacity achieved is not satisfactory. To improve the capacity, the information can be hidden in the video and the audio frames instead of hiding in the Metadata. However, the challenge is the selection of frames, which can resist the compression.

III. Proposed Method

For MP4, typical GOP size is 12. That means every 13th frame is an I-frame. In H.264/AVC, which is the codec for MP4, the approximate quantization parameter for I-frame is 32 and for P and B frames it is 34 and 36 respectively. The theoretical maximum value for any of these frames is 60 [1]. Therefore, only those pixels with values greater than 70 in the I-frame are selected for steganography, so that the secret data is not lost. In addition, the secret information is concealed in duplicates to avoid frame loss. The secret images are buried in the video frames and the secret audio signals are hidden in the audio frames. The embedding and extracting algorithms are given in Fig. 2 and Fig. 3 respectively. The block diagram of embedding procedures is depicted in Fig. 4. The block diagram of the extracting procedure is not shown due to space constraints.

```
Algorithm: Embed- Embeds the secret images and audio in the MP4 video.
Input:
    • Video file (Cover) V.mp4
    • Image (Secret data) im1.jpg and im2.jpg
    • Audio file (Secret data) a.wav.
Output: Video file (Stego) S.mp4.
Method:
1. VidFrd ← vision.VideoFileReader('V.mp4')
2. fid ← VideoReader('V.mp4')
3. p ← fid.NumberOfFrames
4. for j=1 to p do
5.     [Im,A] ← step(VidFrd)
6.     au{j} ← A
7.     i{j}← Im
8. endfor  // Get the video and the audio frames form the input cover video. Keep audio frames in 'au' and video frames in 'i'.
//In every 13th frame, i.e, 1st, 13th, 26th etc., the secret images are hidden three times continuously. The frame is decomposed into Red Green and Blue components. Secret images are hidden in Blue and Green components. From these frames, all pixels with values greater than 70 are used for information hiding//
9. [sLL1, sHL1, sLH1, sHH1] ← lwt2(double(im1), LS)
10. [sLL2, sHL2, sLH2, sHH2] ← lwt2(double(im2), LS) // Obtain IWT of the secret images. LS is the lifting scheme.
11. Hide the approximation coefficients of the secret images im1 and im2 (after encrypting using modified Arnold transformation), three times by replacing fourth, fifth and sixth bits of the pixels of the blue and green components. Also, hide the number of bits of im1 and im2 hidden.
12. Integrate Red, Green, and Blue components to form an image frame and join it to the video in the proper position.
13. Except the first frame, combine all other audio frames.
14. [CAc, CDc] ← lwt(double(frame), LS)
15. [CAs, CDs] ← lwt(double(a), LS) // Apply IWT to the joined audio frames and the input audio 'a' to get approximations and details.
16. Obtain the binary of CAs and duplicate the bits three times.
17. Hide the number of CA coefficients in the first detailed coefficient and the duplicated secret bits in the third, fourth and fifth bit planes of the remaining detailed coefficients of the cover.
18. Apply the inverse IWT to get the stego audio samples and then transform them into frames.
19. Generate the stego video file S.mp4 from the video and the audio frames.
20. return stego video S.mp4
```

Fig. 2. Embedding algorithm

The low-frequency coefficients (also called approximation coefficients) of the transformed secret images are encrypted using modified Arnold transformation. The modified Arnold transformation can be obtained by (1).

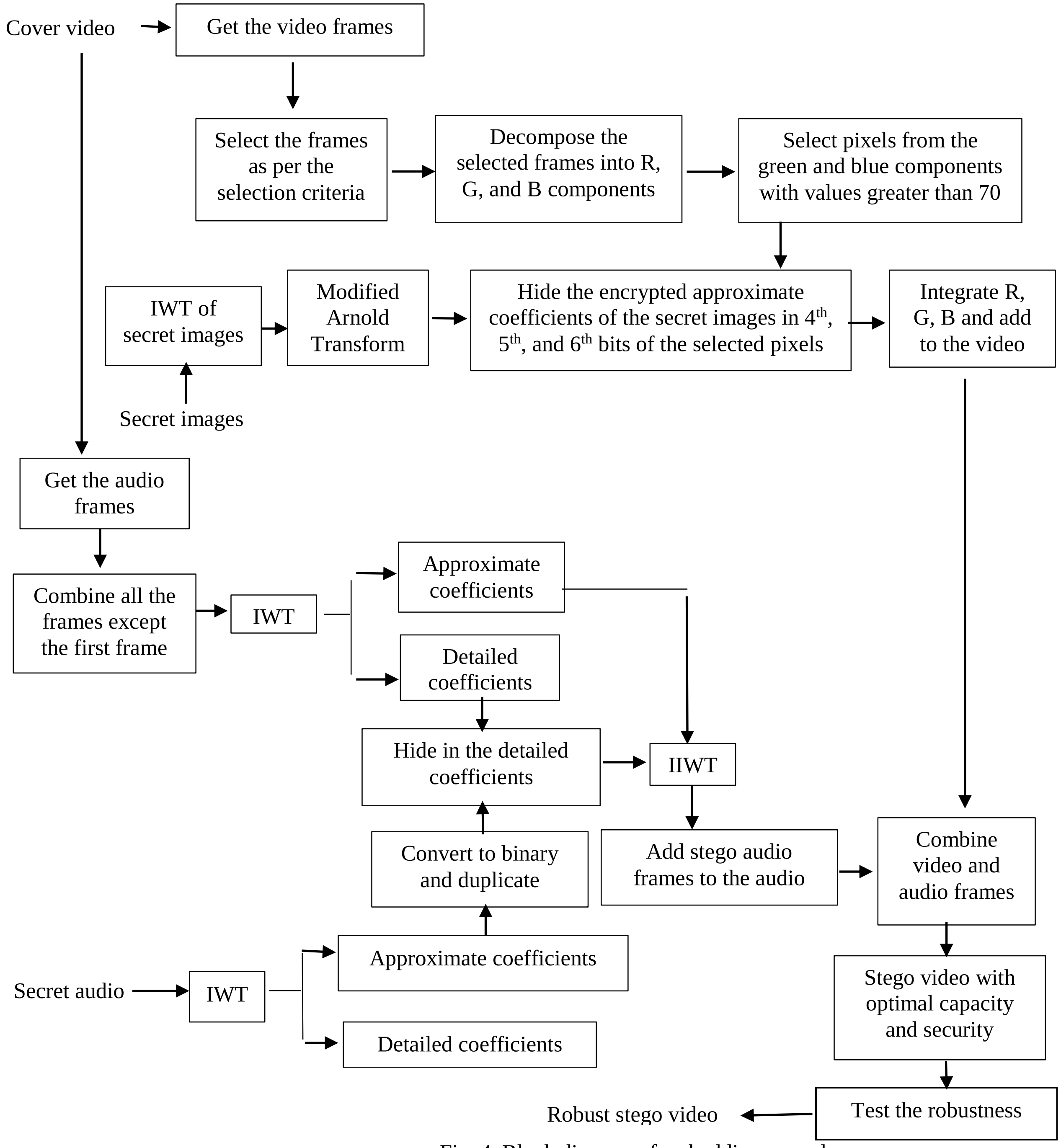
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} i & i+1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} mod\ N \qquad (1).$$

where i ∈{1,2,3...} and N is the size of the digital image. x, y are original pixel coordinates and x', y' are modified pixel coordinates. The key is also hidden in the cover. This process improves the security.

```
Algorithm: Extract- Extracts the secret images and secret audio from the stego video.
Input:  Video file (Stego) S.mp4
Output:
    1. Images im1a.jpg and im2a.jpg
    2. Audio file S3a.wav
Method:
    3. VidFrd←vision.VideoFileReader('S.mp4')
    4. fid ← VideoReader('S.mp4')
    5. p ← fid.NumberOfFrames
    6. for j=1 to p do
    7.     [Im,A] ← step(VidFrd)
    8.     au1{j} ← A
    9.     i1{j} ← Im
    10. endfor  // Get the video and the audio frames form the stego video. Keep audio frames in 'au1' and video frames in 'i1'
    11. Decompose every 1st, 13th, 26th, etc. video frames into Blue and Green components.
    12. Extract the fourth, fifth, sixth bits and obtain the secret coefficients by majority evaluation.
    13. Decrypt using inverse Arnold transformation
    14. im1a ← ilwt2(newSLL1, 0, 0, 0, LS)
    15. im2a ← ilwt2(newSLL2, 0, 0, 0, LS) // Apply the inverse IWT to get secret images. Consider zeros for detail coefficients.
    16. [CAc1, CDc1] ← lwt(double(frame), LS) // Except for the first frame, all other frames are combined and the IWT is applied to get approximate and detail coefficients. First CD coefficient gives the number of secret coefficients hidden.
    17. Extract the third, fourth, and fifth bits of the detail
```

```
     coefficients except the first detail coefficient according
     to the number of secret bits hidden.
18.  Obtain the secret coefficients by majority evaluation.
19.  Apply the inverse IWT to get the secret audio $S3a$.
     Consider zeros for detail coefficients.
20.  **return** *im1a.jpg*, *im2a.jpg*, and *S3a.wav*
```

Fig. 3. Extracting algorithm



Cover video
Get the video frames
Select the frames as per the selection criteria
Decompose the selected frames into R, G, and B components
Select pixels from the green and blue components with values greater than 70
IWT of secret images
Modified Arnold Transform
Hide the encrypted approximate coefficients of the secret images in 4th, 5th, and 6th bits of the selected pixels
Integrate R, G, B and add to the video
Secret images
Get the audio frames
Combine all the frames except the first frame
IWT
Approximate coefficients
Detailed coefficients
Hide in the detailed coefficients
IIWT
Convert to binary and duplicate
Add stego audio frames to the audio
Combine video and audio frames
Approximate coefficients
Secret audio
IWT
Detailed coefficients
Stego video with optimal capacity and security
Test the robustness
Robust stego video

Fig. 4. Block diagram of embedding procedure

A. Experimental Results

MATLAB 8.2 is used for experimentation. Fig. 5 shows one among many frames with size 640 x 360 pixels. Images of Football and Earth shown in Fig. 6 are used as secret images. Secret audio is a music file with 8 bits per sample. Fig. 7 describes the ouputs. Table I shows the performance metrics for the outputs.

The performance of the technique is evaluated using the metrics, Video Quality Metric (VQM) for the stego video, Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Metric (SSIM) for the extracted secret images and Signal to Noise Ratio (SNR) and Squared Pearson Correlation Coefficient (SPCC) for the extracted secret audio.

VQM is computed using the software tool BVQM. PSNR is given by (2).

$$PSNR = 10 \times log_{10}\left(\frac{P^2}{MSE}\right) \qquad (2)$$

where P = maximum pixel value in the original image, MSE = Mean Square Error obtained by (3).

$$MSE = \frac{1}{P\times Q}\sum_{i=1}^{P}\sum_{j=1}^{Q}\|X(i,j) - Y(i,j)\|^2 \qquad (3)$$

Here $X(i,j)$ is an original pixel value, $Y(i,j)$ is stego pixel value, and $P \times Q$ is the dimensions of the image. The PSNR is expressed in decibels (dB).

The SSIM is given by (4).

$$SSIM = \frac{(2\times\bar{x} \times \bar{y}+K1)(2\times\sigma_{xy}+K2)}{({\sigma_x}^2 + {\sigma_y}^2 + K2)\times(\bar{x}^2+\bar{y}^2+K1)} \qquad (4)$$

where K1 = $(c_1N)^2$, and K2 = $(c_2N)^2$ are the two constants. N is the pixel's range (typical value is $2^{\text{number of bits per pixel}}$ -1). By default, c_1 = 0.01 and c_2 = 0.03. The range of SSIM is between -1 and 1. For identical images, it is the maximum value 1. $\bar{x}, \bar{y}, {\sigma_x}^2, {\sigma_y}^2 and\ \sigma_{xy}$ are given as

$$\bar{x} = \frac{1}{P \times Q}\sum_{m=1}^{P}\sum_{n=1}^{Q}(x(m,n))$$

$$\bar{y} = \frac{1}{P \times Q}\sum_{m=1}^{P}\sum_{n=1}^{Q}(y(m,n))$$

$${\sigma_x}^2 = \frac{1}{P \times Q - 1}\sum_{m=1}^{P}\sum_{n=1}^{Q}(x(m,n) - \bar{x})^2$$

$${\sigma_y}^2 = \frac{1}{P \times Q - 1}\sum_{m=1}^{P}\sum_{n=1}^{Q}(y(m,n) - \bar{y})^2$$

$$\sigma_{xy} = \frac{1}{P \times Q - 1}\sum_{m=1}^{P}\sum_{n=1}^{Q}((x(m,n) - \bar{x})\,(y(m,n) - \bar{y}))$$

SNR is given by (5).

$$\text{SNR} = 10\log_{10}\left(\frac{\frac{1}{N}\sum_{i=1}^{N} pi^2}{MSE}\right) \qquad (5)$$

where $MSE(p,q) = \frac{1}{N}\sum_{i=1}^{N}(pi - qi)^2$, p$i$ is the value of the ith sample in original signal, qi is the value of the ith sample in stego signal, N is the number of samples.

SPCC is given by (6)

$$SPCC = \left[\sum\frac{(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2}\sqrt{\sum(y-\bar{y})^2}}\right]^2 \qquad (6)$$

Here,

x: input signal

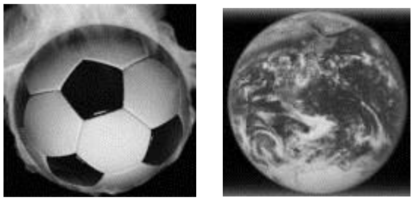
y: output signal

$\bar{x}$: average of the input signal

$\bar{y}$: average of the output signal.

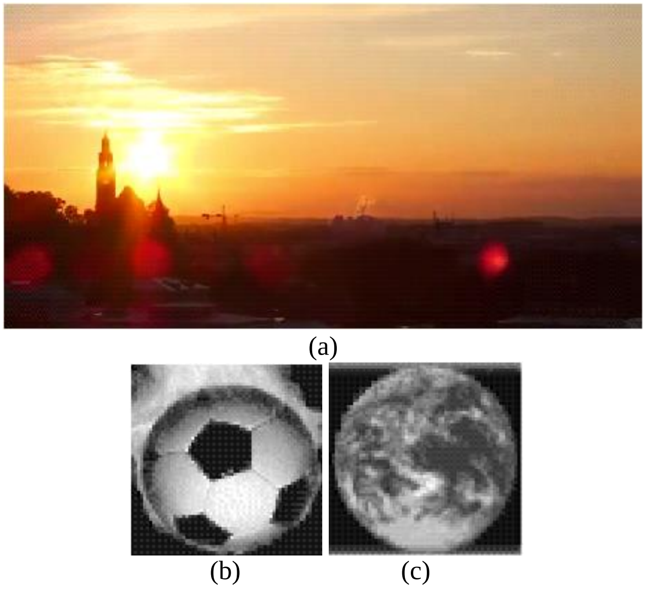
The payload capacity depends on the cover video because only those pixel values or samples with values above the threshold are considered for embedding. C.mp4 is approximately a two-minute video with a frame rate of 30. With this cover video, it is possible to hide up to four images of size 128 x 128 and 32768 audio samples. With this payload, the VQM for the output video is reasonable and the secret information can be extracted without much distortion.

Fig. 5. One of the frames from the video



(a) (b)

Fig. 6. Secret Images: (a) Football image (b) Earth image


(a)

(b) (c)

Fig. 7. Output frame and Retrieved images: (a) output frame (b) retrieved Football image (c) retrieved Earth image

Table I. Performance metrics for MP4 video steganography

Cover video	Secret images (grayscale) and Secret audio	Stego	Extracted image		Extracted audio	
		Average VQM	PSNR in dB	SSIM	SNR in dB	SPCC
C.mp4	Image size: 128 x 128. No. of images: 2 No. of audio samples: 20000	0.0003	im1:31 im2:32	im1:0.9023 im2:0.9565	29	0.8235
	Image size: 128 x 128 No. of images: 4 No. of audio samples: 32768	0.0052	im1:30 .5 im2:29 im3:30 im4:28	im1:0.9015 im2:0.8953 im3:0.9025 im4:0.8825	25	0.7906

B. Performance Against Attacks

Table II shows performance against RST (Rotation, Scaling, and Translation) attack. The secret image and the secret audio can be extracted without distortion. All the stego frames are rotated by two degrees and then scaled. The secret information is extracted with reasonable performance metrics. If the rotation is more than this, the secret data cannot be retrieved.

Table II. Performance against RST attack

Cover video	Secret images (grayscale)	Stego	Extracted image		Extracted audio	
		VQM	PSNR in dB	SSIM	SNR in dB	SPCC
C. mp 4	Image size: 128 x 128. No. of images: 2 No. of audio samples: 20000	0.0195	im1:27 im2:28	im1: 0.7985 im2: 0.8295	27	0.7865

Table III shows the comparison of proposed video steganography technique with S. Po-Chyi et al.'s technique [11] that hides the secret information in both the video and the audio frames of an FLV video. But the performance of the audio is not analyzed in this paper and hence only PSNR is compared. The proposed technique, in this case, improves the capacity by 4.5% with a decrease in PSNR by 5dB.

Table III. Comparison with related published work

Metrics	S. Po-Chyi et al. [11]	Proposed
Payload capacity in %	13.7	18.2
PSNR in dB	36	31

IV. CONCLUSION

A strong, high capability video steganography method is proposed in the paper. Secret images and audio are veiled in the MP4 compressed video. I-frames are used for embedding the images which increase the capacity, the security, and the robustness. The secret information is not hidden in the Metadata as found in the literature, in which case the capacity is very low. Secret audio is hidden in the audio frames. Very few papers are found where both image and audio are buried in the video. Significant results are obtained compared to the existing work

V. REFERENCES

[1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), AVC Reference Manual, 2009.

[2] S. Hemalatha, A. U. Dinesh and A. Renuka, "Audio data hiding technique using integer wavelet transform," *International Journal of Electronic Security and Digital Forensics,* vol. 8, no. 2, pp. 131-146, 2016.

[3] S. Hemalatha, A. U. Dinesh, A. Renuka and K. R. Priya, "A SECURE AND HIGH CAPACITY IMAGE STeganography Technique," *Signal & Image Processing : An International Journal,* vol. 4, no. 1, pp. 83-89, 2013.

[4] G. Elham, S. Jamshid and Z. Bahram, "A Steganographic Method based on Integer Wavelet Transform and Genetic Algorithm," in *Proceedings of the IEEE International*

Conference on Communications and Signal Processing, 2011.

[5] N. Hideki, F. Tomonori, N. Michiharu and K. Eiji, "Application of BPCS Steganography to Wavelet Compressed Video," in *Proceedings of International Conference on Image Processing*, 2004.

[6] X. Changyong, P. Xijian and Z. Tao, "Steganography in Compressed Video Stream," in *Proceedings of First International Conference on Innovative Computing, Information and Control*, 2006.

[7] A. J. Mozo, M. E. Obien, C. J. Rigor, D. F. Rayel, K. Chua and G. Tangonan, "Video Steganography using Flash Video (FLV)," in *Proceedings of IEEE Conference on Instrumentation and Measurement Technology*, 2009.

[8] P. Feng, X. Li, Y. Xiao-Yuan and G. Yao, "Video Steganography using Motion Vector and Linear Block Codes," in *Proceedings of IEEE International Conference on Software Engineering and Service Sciences*, 2010.

[9] A. A. Hussein, "Data Hiding in Motion Vectors of Compressed Video Based on Their Associated Prediction Error," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 14-18, 2011.

[10] S. Zafar and W. P. Marc Chaumont, "Considering the Reconstruction Loop for Data Hiding of Intra- and Inter-Frames of H.264/AVC," *Signal, Image and Video Processing*, vol. 7, no. 1, pp. 75-93, 2013.

[11] S. Po-Chyi, L. Ming-Tse and W. Ching-Yu, "A Practical Design of High-Volume Steganography in Digital Video Files," *Multimedia Tools and Applications*, vol. 66, p. 247–266., 2013.

[12] G. Sagar and B. B. Amberker, "DCT Based Reversible Data Embedding for MPEG-4 Video using HVS Characteristics," *Journal of information security and applications*, vol. 18, pp. 157 -166, 2013.

[13] L. Yunxia, L. Zhitang, M. Xiaojing and L. Jian, "A Robust Data Hiding Algorithm for H.264/AVC Video Streams," *The Journal of Systems and Software*, vol. 86, no. 8, pp. 2174-2183, 2013.

[14] T. Shanableh, "Data Hiding in MPEG Video Files Using Multivariate Regression and Flexible Macroblock Ordering," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 7, no. 2, pp. 455-464, 2012.

[15] D. Xu, R. Wang and Q. Shi Yun, "Data Hiding in Encrypted H.264/AVC Video Streams by Codeword Substitution," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 596-606, 2014.

[16] Y. Yuanzhi, Z. Weiming and Y. Nenghai, "Defining Embedding Distortion for Motion Vector-Based Video Steganography," *Multimedia tools and Applications*, pp. 1-24, 3 September 2014.

[17] S. Hemalatha, A. U. Dinesh and A. Renuka, "High Capacity Video Steganography Technique in Transform Domain," *Pertanika Journal of Science and Technology*, vol. 24, no. 2, pp. 411- 422, 2016.