International Journal of Engineering Technology and Applied Science
(ISSN: 2395 3853), Vol. 2 Issue 11 November 2016

Video Steganography Based on Compression and Embedding Methodology using Spatial and Transform Domain

[1]Pratiksha Singh, [2]Sneha Singh, [3]Sushil Chaturvedi
[1]Mtech-Scholar, [2]Guide & HOD, [3]Co-Guide & Assistant Professor
[123]JNCT, REWA, INDIA

Abstract
This work proposes a novel method of Steganography in the Video domain which is a largely unexplored area. The mechanism uses a hybrid approach and makes use of Singular Value Decomposition and Discrete Wavelet Transform to achieve the desired objective. The algorithm has been rigorously tested on an extended database of videos using embedding strength as a variable. The results have been evaluated on the basis of visual imperceptibility, PSNR and MSE of the extracted cover image. It has been found that the mechanism is able to achieve a highly imperceptible Stego Video from which the cover image can be extracted at the receiver without any prior knowledge of the Original cover image, therefore establishing the idea of a fully blind mechanism. The metrics of PSNR, MSE and Correlation factor further augment the statement that the mechanism achieves the desired objectives of imperceptibility along with faithful recovery of the concealed secret image.

KEYWORDS: Steganalysis, Decomposition, DCT, DWT, Stegnography

I INTRODUCTION

Steganography means "covered writing". It is defined as the art of hiding information in ways that prevent the detection of hidden messages [1]. At the beginning, we briefly introduce the terminology used throughout the paper. The term "cover object" describes the file used for hiding information. The "secret message" refers to the data that is embedded in the cover through an embedding module. A "stego-object" is produced combining the cover object with the embedded data. In case of encrypting the secret message before embedding, an encryption key is used. This key is referred to as "stego-key". Furthermore, the term "steganalysis" refers to the different attacks that try to break the steganographic algorithm. Figure 1 shows a general steganographic model.

The design of a good steganographic technique faces many challenges. The algorithm's computational complexity and whether the algorithm is blind [2, 3, 4, 5] or non-blind should be considered. Unfortunately, most of the existing algorithms do not discuss their computational complexity. Mainly, there are four challenges: robustness, tamper resistance, hiding capacity and perceptual transparency. All of these aspects are inversely proportional to each other creating the data hiding dilemma. Robustness is the amount of modification the stego-object could withstand before an adversary destroys the hidden information [6]. While tamper resistance is the difficulty for an attacker to change the secret message after it has been embedded in the cover object. On the other hand, there is a trade-off between the hiding capacity and the perceptual transparency. When the hiding capacity increases, a smaller cover object could be used for hiding the secret message. This results a stego-object with a smaller size that can be easily transmitted over the internet. But increasing the hiding capacity leads to distortions in the stego-object. If an attacker recognizes the distortion, then the presence of the hidden message is detected. And at that point, steganography has failed as the secret communication was revealed.

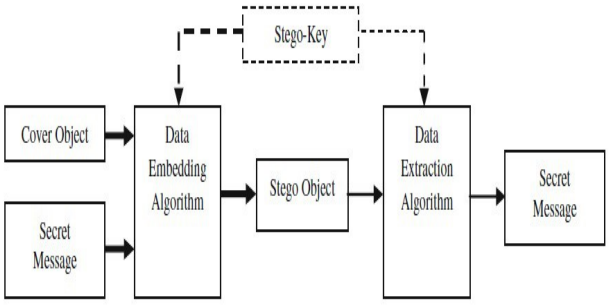

Figure 1 General steganographic model. Embedding process is represented with bold arrows, while extraction process is represented with non-bold arrows.

II PERFORMANCE MEASURES

The growth of video steganography raised a need for computational methods to evaluate the visual quality of the stego–video in a step towards evaluating the steganographic technique itself. Obviously, secret data embedding by any steganographic technique changes the quality of the original cover ranging from a slight alteration which is not noticeable by the human eye, to clear distortion that can be detected easily. Deciding whether the steganographic technique is perceptually transparent or not, needs

standard metrics that can measure the alteration to the perceptual layout of the stego video. Basically, these metrics are used as an approximation to the human perception of stego-video quality.

The human vision test is the first type of measurement that has been found to measure the quality of steganographic objects after embedding the hidden data [25]. The human vision test is done with the naked eye, by surveying some persons who are asked to examine the original and the stego videos, then they report if they noticed any distortion or alteration in the perceptual vision of the stego video by providing a score. This score is known as Mean Opinion Score (MOS) [70]. However, it is highly subjective and hence is inefficient, not reliable, and expensive in terms of preparation time, running and human resources needed. As a result, other objective quality metrics evolved.

Objective quality metrics are algorithms that can numerically estimate the quality of the video, thus predicting the viewers MOS. These metrics can be divided into two different approaches: error-based approach and structural distortion approach. Error-based approach depends on the fact that there is a direct relation between the decrease in perceptual quality and the visibility of the error signal so it measures different aspects of the error signal depending on their visibility to the human eye. On the other hand, structural distortion approaches measures the distortion by using the structural distortion in the video. It is based on the assumption that HVS focuses on extracting the structural information from the viewing field. As a result, the distortion increases as the structural information decreases. In this case, the distortion is not related to the errors.s

In this section, we will review mean square error-based metrics, video quality metric and moving pictures quality metric, as examples of error-based metrics. While Structural Similarity Index and General Video Quality Model, are reviewed as examples of structural distortion metrics.

Mean Square Error (MSE) represents the accumulated squared error between original and stego-frames. The square root of the MSE represents the Root Mean Square Error (RMSE). A very close metric is the Signal -to-Noise Ratio (SNR) which measures the amount of noise corruption to the original signal. Similarly, the Peak Signal-to-Noise Ratio (PSNR) calculates the highest SNR between two images/frames. As the PSNR increases, the quality of the stego object is better. An enhancement to PSNR is the Weighted PSNR (WPSNR). It considers that the human eye is less sensitive to changes in textured areas than in smooth areas. Notice that, all of the above metrics are measured in decibel (dB). The formulas used to calculate these metrics are provided. Because of their simplicity and low computational complexity, PSNR and MSE are widely used as quality metrics. Because of their simplicity and low computational complexity, PSNR and MSE are widely used as quality metrics. The Peak-Signal-To-Noise Ratio (PSNR) is used to measure deviation of the watermarked and attacked frames from the original video frames and is defined as:

$PSNR = 10 \log_{10} 255^2/MSE$

Where MSE (mean squared error) between the original and distorted frames (size m x n) is defined as:

$$MSE = (1/mn) \sum_{i=1}^{m} \sum_{j=1}^{n} [I(i,j) - I'(i,j)]$$

III PROPOSED METHODOLOGY

This section elaborates the embedding of secret image into original video and extraction of cover image from stego video. The steps for steganograpy procedure are as follows:

1. The original video is partitioned into groups of P frames
2. R, G, B planes are isolated from each frame of the original video.
3. Each plane of the frame is decomposed using DWT into bands of different frequencies.
4. The DCT is performed on selected high and middle frequency bands to get DCT transformed coefficients.
5. Singular values are obtained by applying SVD to DCT coefficients.
6. R, G, B planes of individual watermarks also separated.
7. DWT is applied on each plane of the individual watermarks to decompose into different frequency bands.
8. DCT is applied on middle and high frequency bands.
9. SVD is performed on DCT coefficients to obtain singular values.
10. Now the singular values of R, G, B planes of each frame in video are modified with singular values of R, G, B planes of individual watermarks at chosen scaling factor. Stego frame is obtained by using the following :

$M(i, j) = So(i, j) + \alpha * Sw(i, j)$

11. Inverse SVD is applied to get modified matrix using orthogonal matrices.
12. Inverse DCT is performed on modified SVD matrices.
13. Stego frame is attained by applying Inverse DWT to coefficients obtained in step 12.

14. These steps are repeated for embedding the individual watermarks in each group of P frames.

15. PSNR values are estimated between watermarked and original video frames.

With this algorithm, we are able to embed the maximum capacity of Steganography without degrading the quality of Stego video.

B. Cover Image detection The extraction process requires the original data to detect the watermark. The extraction steps are as follows:

1. The Stego video is divided into groups of P frames.

2. R, G, B planes are separated from stegoframe of video.

3. DWT is applied to R, G, B planes of each video frame to get decomposed frequency bands.

4. High energy compaction coefficients are acquired by performing DCT on selected frequency bands.

5. Modified singular values are secured by applying SVD on DCT coefficients.

6. Singular values of extracted image is obtained by using the following:

$Sw(i, j) = (M(i, j) – S0(i, j)) / \alpha$

7. Inverse SVD is applied by using orthogonal matrices of original Cover Image.

8. Inverse DCT is performed.

9. Secret Data/Cover Image is extracted by applying inverse DWT on all approximation and transformed detailed coefficients. .

10. Correlation factors are estimated between extracted and embedded stego images

IV FUNCTIONS USED FOR VIDEO PROCESSING

1) **VideoReader** class: Read video files

Description: Use the Video Reader function with the read method to read video data from a file into the MATLAB workspace. The file formats that Video Reader supports vary by platform, as follows (with no restrictions on file extensions): It supports all Platforms AVI (.avi), Motion JPEG 2000 (.mj2) All Windows MPEG-1 (.mpg), Windows Media Video (.wmv, .asf, .asx), and any format supported by Microsoft DirectShow®. Windows 7 MPEG-4, including H.264 encoded video (.mp4, .m4v), Apple QuickTime Movie (.mov), and any format supported by Microsoft Media Foundation.

Use and Construction

obj = VideoReader(filename) constructs obj to read video data from the file named filename. If it cannot construct the object for any reason, VideoReader generates an error.

obj = VideoReader(filename,Name,Value) constructs the object with additional options specified by one or more Name,Value pair arguments. Name is 'Tag' or 'UserData' and Value is the corresponding value. You can specify two name and value pair arguments in any order as Name1,Value1,Name2,Value2.

Input Arguments

1) Filename: String in single quotation marks that specifies the video file to read. The VideoReader constructor searches for the file on the MATLAB path.
2) Name-Value Pair Arguments: Optional comma-separated pairs of Name,Value arguments, where Name is 'Tag' or 'UserData' and Value is the corresponding value. You can specify two name and value pair arguments in any order as Name1,Value1,Name2,Value2.
3) 'Tag': String that identifies the object.
4) 'UserData': Generic field for data of any class that you want to add to the object.

Properties:

All properties are read only except Tag and UserData.

2**) imread :** Read image from graphics file

Syntax

A = imread(filename, fmt)
[X, map] = imread(...)
[...] = imread(filename)
[...] = imread(URL,...)
[...] = imread(...,Param1,Val1,Param2,Val2...)

Description

A = imread(filename, fmt) reads a grayscale or color image from the file specified by the string filename. If the file is not in the current folder, or in a folder on the MATLAB path, specify the full pathname. The text string fmt specifies the format of the file by its standard file extension. For example, specify 'gif' for Graphics Interchange Format files. To see a list of supported formats, with their file extensions, use the imformats function. If imread cannot find a file named filename, it looks for a file named filename.fmt. The return value A is an array containing the image data. If the file contains a grayscale image, A is an M-by-N array. If the file contains a truecolor image, A is an M-by-N-by-3 array. For TIFF files containing color images that use the CMYK color space, A is an M-by-N-by-4 array. See TIFF in the Format-Specific Information section for more information. The class of A depends on the bits-per-sample of the image data, rounded to the next byte boundary. For example, imread returns 24-bit color data as an array of uint8 data because the sample size for each color component is 8 bits. See Tips for a discussion of bitdepths, and see Format-Specific Information for more detail about supported bit depths and sample sizes for a particular format.

3) **im2bw**: Convert image to binary image, based on threshold

Syntax

BW = im2bw(I, level)
BW = im2bw(X, map, level)
BW = im2bw(RGB, level)
Description
BW = im2bw(I, level) converts the grayscale image I to a binary image. The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). Specify level in the range [0,1]. This range is relative to the signal levels possible for the image's class. Therefore, a level value of 0.5 is midway between black and white, regardless of class. To compute the level argument, you can use the function graythresh. If you do not specify level, im2bw uses the value 0.5.
BW = im2bw(X, map, level) converts the indexed image X with colormap map to a binary image.
BW = im2bw(RGB, level) converts the truecolor image RGB to a binary image.

4) **Implay**: Play movies, videos, or image sequences

Syntax
Implay
implay(filename)
implay(I)
implay(..., FPS)
Description
implay opens a Movie Player for showing MATLAB movies, videos, or image sequences (also called image stacks). Use the implay File menu to select the movie or image sequence that you want to play. Use implay toolbar buttons or menu options to play the movie, jump to a specific frame in the sequence, change the frame rate of the display, or perform other exploration activities. You can open multiple implay movie players to view different movies simultaneously.
The following figure shows the Movie Player containing an image sequence.
implay(filename) opens the implay movie player, displaying the content of the file specified by filename. The file can be an Audio Video Interleaved (AVI) file. implay reads one frame at a time, conserving memory during playback. implay does not play audio tracks.
implay(I) opens the implay movie player, displaying the first frame in the multiframe image array specified by I. I can be a MATLAB movie structure, or a sequence of binary, grayscale, or truecolor images. A binary or grayscale image sequence can be an M-by-N-by-1-by-K array or an M-by-N-by-K array. A truecolor image sequence must be an M-by-N-by-3-by-K array.If the input image is not a grayscale image, im2bw converts the input image to grayscale, and then converts this grayscale image to binary by thresholding.

5) **dwt2 :**Single-level discrete 2-D wavelet transform

Syntax
[cA,cH,cV,cD] = dwt2(X,'wname')
[cA,cH,cV,cD] = dwt2(X,Lo_D,Hi_D)
Description
The dwt2 command performs a single-level two-dimensional wavelet decomposition with respect to either a particular wavelet ('wname', see wfilters for more information) or particular wavelet decomposition filters (Lo_D and Hi_D) you specify.
[cA,cH,cV,cD] = dwt2(X,'wname') computes the approximation coefficients matrix cA and details coefficients matrices cH, cV, and cD (horizontal, vertical, and diagonal, respectively), obtained by wavelet decomposition of the input matrix X. The 'wname' string contains the wavelet name.
[cA,cH,cV,cD] = dwt2(X,Lo_D,Hi_D) computes the two-dimensional wavelet decomposition as above, based on wavelet decomposition filters that you specify.
Lo_D is the decomposition low-pass filter.
Hi_D is the decomposition high-pass filter.
Lo_D and Hi_D must be the same length.
Let sx = size(X) and lf = the length of filters; then size(cA) = size(cH) = size(cV) = size(cD) = sa where sa = ceil(sx/2), if the DWT extension mode is set to periodization. For the other extension modes, sa = floor((sx+lf-1)/2).
[cA,cH,cV,cD] = dwt2(...,'mode',MODE) computes the wavelet decomposition with the extension mode MODE that you specify.

6) svd: Singular value decomposition

Syntax
s = svd(X)
[U,S,V] = svd(X)
[U,S,V] = svd(X,0)
[U,S,V] = svd(X,'econ')
Description
The svd command computes the matrix singular value decomposition.
s = svd(X) returns a vector of singular values.
[U,S,V] = svd(X) produces a diagonal matrix S of the same dimension as X, with nonnegative diagonal elements in decreasing order, and unitary matrices U and V so that X = U*S*V'.
[U,S,V] = svd(X,0) produces the "economy size" decomposition. If X is m-by-n with m > n, then svd computes only the first n columns of U and S is n-by-n.
[U,S,V] = svd(X,'econ') also produces the "economy size" decomposition. If X is m-by-n with m >= n, it is

equivalent to svd(X,0). For m < n, only the first m columns of V are computed and S is m-by-m.

Because of their simplicity and low computational complexity, PSNR and MSE are widely used as quality metrics. The Peak-Signal-To-Noise Ratio (PSNR) is used to measure deviation of the watermarked and attacked frames from the original video frames and is defined as:

$PSNR = 10 \log_{10} 255^2/MSE$

Where MSE (mean squared error) between the original and distorted frames (size m x n) is defined as:

$$MSE = (1/mn) \sum_{i=1}^{m} \sum_{j=1}^{n} [I(i,j) - I'(i,j)]$$

V DATABASE SPECIFICATIONS

The sample data has been taken from the standard Matlab database of the Computer Vision Toolbox. Computer Vision System Toolbox provides algorithms and tools for video processing. You can read and write from common video formats, apply common video processing algorithms such as deinterlacing and chroma-resampling, and display results with text and graphics burnt in to the video. Video processing in MATLAB uses System objects™, which avoids excessive memory use by streaming data for processing one frame at a time. The list of files used in the testing are

1) Viptrain2
   Size(Frame width x Frame Height): 352x 288
   Frame Rate: 25 fps
   Total bit rate: 1374 kbps
2) Viptrain
   Size(Frame width x Frame Height): 360x 240
   Frame Rate: 30 fps
   Total bit rate: 4000 kbps

VI MATLAB RESULTS

The results have been given in the below mentioned figures. Figure 2 shows the working environment screen shot. figures 3 and 4 show the stego and the extracted cover image of the logo of the college along with the original image.

The metrics used for the evaluation of performance of our technique are computation of PSNR and MSE of the extracted Cover image in reference to the original hidden image. Embedding strength have been used as a variable. Increase or decrease in the value of x i.e Embedding Strength to improve or weaken the embedding strength. Higher x values will be more tamper proof, however, will result in a larger loss in SNR. Embedding is done on the Y frame only. Figure 1 shows the Matlab environment in which the computations have been done and which generates the Stego Video and extracts the Hidden image as well.

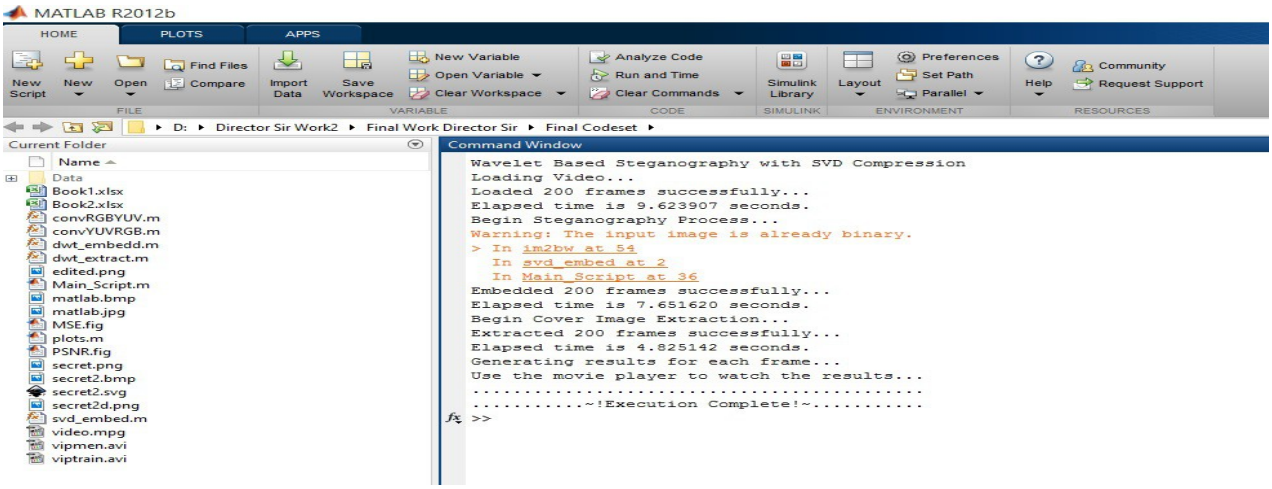

Figure2 Computation Environment

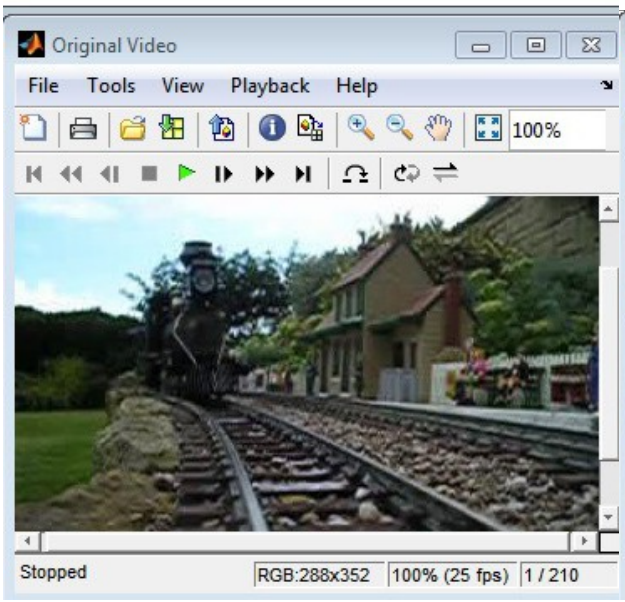

Figure 3 Original Video viptrain

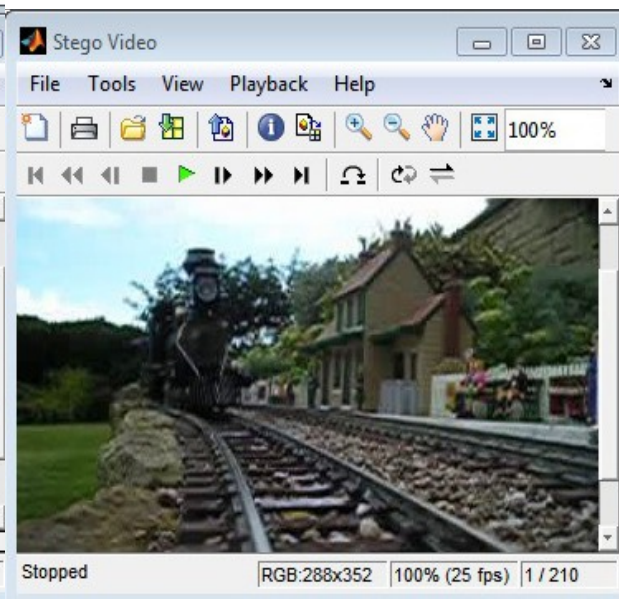

Figure 4 Stego Video viptrain

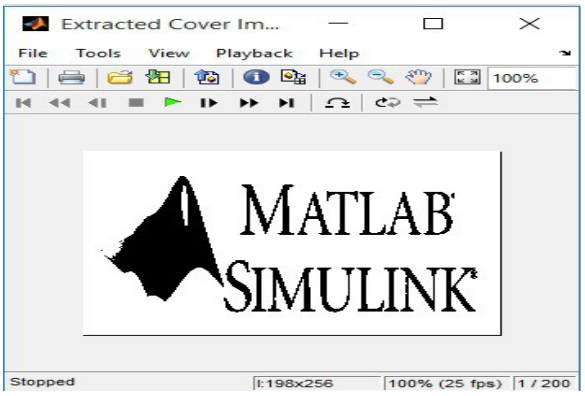

Figure 5 Extracted Cover Image with x =0

Extraction is done using the Y frame of the YUV converted stego video. This frame is already saved in the Yf_out variable and therefore used directly. A similar result can be observed by converting RGB variable frames one by one into YUV and taking the Y frame as the embedded frame. Similarly, the original Y frame is stored in the Yorg variable. Those two are used as inputs to the extractor. The cover image itself is not used at all therefore making it a completely blind.

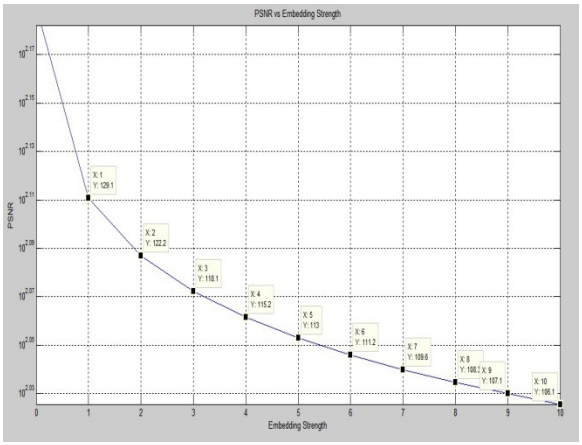

Figure 6PSNR Vs Embedding Strength

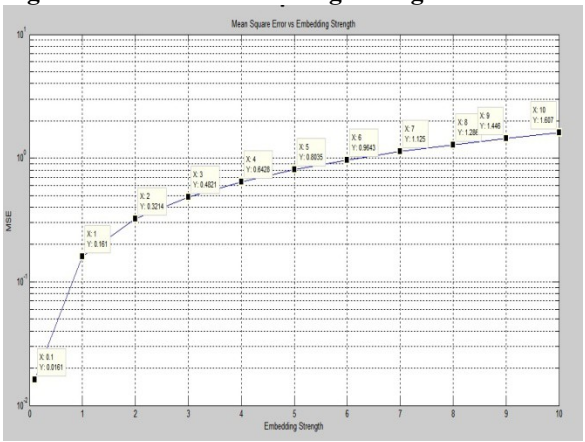

Figure 7 MSE vs Embedding Strength

PSNR values with low values of embedding strength, will be high values. Higher PSNR suggests a higher

imperceptibility. Which means that, if x is decreased from 10 to 0.1, PSNR will increase from ~118 to ~165. However, when Yf_out is tampered with (attacked), PSNR will be lower because the MSE will increase.

Normalized Correlation parameter has been computed which compares the embedded cover image with extracted cover image and has been found to be of value equal to 1 will always be = 1 when no attack has occurred. However, it is bound to degenerate when Yf_out is tampered i.e on an attacked or rotated video.

CONCLUSION

The work started with an objective of realization of a highly imperceptible steganographic system in the Video domain with sufficiently large data hiding capacity. The work started with a comprehensive review of video steganographic techniques. Difference between steganography, cryptography, and watermarking were discussed. An overview of steganography using different cover types was presented and special attention was paid to video steganography and its applications. Various categorizations of the existing techniques were illustrated. The analysis of the state of the art in the field of Video Steganography helped us formulate the problems in this work and set the objectives to be achieved at the end of this work.

REFERENCES

1. (2008) Objective Perceptual Multimedia Video Quality Measurement in the Presence of a Full-Reference, ITU-T Rec. J. 247

2. Abbass AS, Soleit EA, Ghoniemy SA (2007) Blind video data hiding using integer wavelet transforms. Ubiquit Computer Communication J 2(1)

3. Ahsan K, Kundur D (2002) Practical data hiding in TCP/IP. In: Proc. Of Workshop on Multimedia Security bat ACM Multimedia

4. Alattar AM, Alattar OM (2004) Watermarking electronic text documents containing justified paragraphs band irregular line spacing. In: Proc. of SPIE 685–695

5. Al-Frajat AK, Jalab HA, Kasirun ZM, Zaidan AA, Zaidan BB (2010) Hiding data in video file: anboverview. J of Appl Sci (Faisalabad) 10(15):1644–1649

6. Anderson RJ, Petitcolas FAP (1998) on the limits of steganography. IEEE J Sel Areas Commun 16(4): 474–481

7. Bailey K, Curran K (2006) an evaluation of image based steganography methods. Multimed Tools Appl 30(1):55–88

8. Balaji R, Naveen G (2011) secure data transmission using video Steganography. In: IEEE International Conference on Electro/Information Technology (EIT) 1–5

9. Calderbank AR, Daubechies I, Sweldens W, Yeo B-L (1997) Lossless image compression using integer to integer wavelet transforms. In: Proceedings of International Conference on Image Processing 596–599

10. Carli M, Campisi P, Neri (2006) A Data hiding driven by perceptual features for secure communications. In: International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICN/ICONS/MCL) 85–85.

11. Chae JJ, Manjunath BS (1999) Data hiding in video. In: Proceedings of International Conference on Image Processing (ICIP 99) 311–315

12. Chandramouli R, Memon ND (2003) Steganography capacity: A steganalysis perspective. In: Proceedings of SPIE 173–177

13. Chang K-C, Chang C-P, Huang PS, Tu T-M (2008) A novel image steganographic method using tri-way pixel-value differencing. J Multimedia 3(2):37–44

14. Chang F-C, Hang H-M, Huang H-C (2007) Layered access control schemes on watermarked scalable media. J VLSI Signal Process System Signal Image Video Technology 49(3):443–455

15. Channalli S, Jadhav A (2009) Steganography an Art of hiding data. Int J Computer Science Engineering (IJCSE) 1(3): 137–141

16. Cheddad A, Condell J, Curran K, Mc Kevitt P (2009) A skin tone detection algorithm for an adaptive approach to steganography. Signal Process 89(12):2465–2478

17. Cheddad A, Condell J, Curran K, Mc Kevitt P (2010) Digital image steganography: survey and analysis of current methods. Signal Process 90(3):727–752

18. Das R, Tuithung T (2012) a novel steganography method for image based on Huffman Encoding. In: 3rd National Conference on Emerging Trends and Applications in Computer Science (NCETACS) 14–18

19. Eltahir ME, Kiah LM, Zaidan BB, Zaidan AA (2009) High rate video streaming steganography. In: International Conference on Future Computer and Communication (ICFCC 2009) 672–675

20. Fridrich J, GoljanM, Du R (2001) Detecting LSB steganography in color, and gray-scale images. Multimedia IEEE 8(4):22–28

21. Hamid N, Yahya A, Ahmad RB, Al-Qershi OM(2012) Image steganography techniques: an overview. Int J Computer Science Security (IJCSS) 6(3):p168–p187

22. Hanafy AA, Salama GI, Mohasseb YZ (2008) a secure covert communication model based on video

steganography. In: Military Communications Conference (MILCOM 2008) 1–6.