Hiding Large Amount of Data using a New Approach of Video Steganography

Conference Paper · September 2013

DOI: 10.1049/cp.2013.2338

1 author:

Virendra Yadav
ABES Engineering College

Hiding Large Amount of Data using a New Approach of Video Steganography

Rahul Paul[1], Anuja Kumar Acharya[2],
Virendra Kumar Yadav[3], Saumya Batham[4]

[1]School of Computer Engineering, KIIT University, Bhubaneswar, India
[1]rahulpaul123@gmail.com, [2]anuja.signin@gmail.com,
[3]virendrashines@gmail.com, [4]saumyabatham003@gmail.com

ABSTRACT

Steganography has become one of the most robust technique to transmit secrete or sensitive information between parties by concealing the information in some file which may appear to be of no importance to any interceptor. In this paper, we have used video file as the cover file and embedded the information in the frames where the scene has changed in the video sequence using the Least Significant Bit (LSB) replacement technique. For additional security we have randomized the pixel positions where the information bits are embedded by generating an indexed based chaotic sequence and arranging the pixel position according to the sequence. The experimental result shows that the original cover video and stego video are visually identical i.e. there is no perceptual difference between the two videos.

KEYWORDS

Steganography, cover video, scene change, Least Significant Bit (LSB), randomization, Indexed based chaotic sequence.

1. INTRODUCTION

Steganography has become one of the most robust and efficient technique to send secrete or sensitive information to another party without the knowledge of any interceptor. In steganography the sender uses any ordinary file like a video, image, audio, etc. known as carrier or cover file which would appear to be of no importance to any interceptor. He then transmits the information or data by embedding it in the cover file using various techniques. Steganography and cryptography are used to transmit secrete messages or information. One of the major differences between these two techniques is that, in cryptography the information to be transmitted is transformed into some other form that cannot be understood by any unauthorized party, whereas in steganography the information is concealed or embedded into some other file may it be a video, image, audio etc. which appears to be of no importance to an interceptor. Thus steganography has an added advantage over cryptography i.e. in cryptography since the information is in a visible encrypted form it arouse suspicion and attracts various types of attack but in steganography the information is hidden behind some other files and therefore arouse less or no suspicion and therefore attracts less type of attacks. This advantage is due to the human nature by which, the human mind generally ignores all such files that appears to be of no importance to him and thus does not consider if there is some hidden information inside that file unless he gets some clue. In case of cryptography that clue is the visible encrypted form of the information but in case of steganography no such clue is available.

With the increase in the amount of information transmitted in recent times, it has become important to select cover files such that they can hold a large volume of information. Therefore in this paper we have focused on video files as cover files because a video file can contain a large amount of information as compared to any image, audio or document file.

2. PREVIOUS WORK

Paper [1] has described the various techniques of Steganography. The techniques can be broadly classified into five major categories. These are the spatial – domain based steganography that includes the LSB replacement and matching and BPCS techniques, transform – domain based steganography which directly computes over the transform coefficients, document based steganography that includes embedding the information in the documents, file structure based steganography in which the structure of the cover file is exploited to embed the information and other techniques that used video compression encoding scheme and spread spectrum technique. Chi-Kwong Chan and L.M. Cheng [2] proposed to apply an optimal pixel adjustment process over a simple LSB substitution method to improve the visual image quality of the stego image as compared to cover image. Dr. V. Vijayalakshmi, Dr. G. Zayaraz and V. Nagaraj [3] proposed a modulo – based LSB Steganography algorithm taking image as cover image to provide more protection against image steganalysis based on histogram analysis and statistical analysis. Kazem Ghazanfari, Shahrokh Ghaemmaghami and Saeed R. Khosravi [4] proposed an improvement over the LSB^{+}

method that embeds some extra bits to make the histogram of the cover image and stego image look similar to resist histogram based image steganalysis by reducing the amount of visual and statistical characteristics of the cover image. Yi-Chun Liao, Chung-Han Chen, Timothy K.Shih and Nick C. Tang [5] proposed a new video steganography technique by applying the adaptive LSB data hiding technique for image over the video and using a 3 dimensional array to store the information about the pixels of the cover image and stego image. Mohamed Elsadig, Miss Laiha Mat Kiah, Bilal Bahaa Zaidan and Aos Alaa Zaidan [6] proposed a video steganography algorithm using the LSB replacement technique and embedding the message bits in a 3-3-2 approach i.e. 3 message bits are embedded in the red plain, 3 bits are embedded in green plain and 2 bits are embedded in the blue plain of the cover image.

The rest of the paper is organized in the following manner. Section 3 briefly states what scene change detection is and what techniques are used for this. Section 4 describes the Indexed based chaotic sequence. Section 5 describes the proposed model. Section 6 describes the data capacity. Section 7 describes the experimental result and performance analysis. Finally section 8 gives the conclusion.

3. SCENE CHANGE DETECTION

Paper [7] gives a brief introduction to what is scene change detection and the techniques used to detect the scene change in a video sequence. A video consists of a sequence of frames. Every frame is not the same. Any change between the current frame and the previous frame indicates a change of scene. Scene changes take place either gradually or abruptly. A gradual change occurs due to changes in the chromatic features, spatial features or both and an abrupt change occurs due to editing. There are different techniques that are used in scene change detection. These techniques are – pixel difference, sum of absolute difference (SAD), statistical difference, block difference, histogram difference, edge tracking and using macroblocks to detect the point of change of scene.

4. INDEXED BASED CHAOTIC SEQUENCE

Chaotic sequence function is a random sequence generator function that is used to generate pseudorandom sequence using the following equation:

$$x_{n+1} = \mu x_n(1 - x_n) \quad (1)$$

Equation (1) produces a random sequence of real numbers. But for deciding the pixel position we require integers. The real value sequence can be converted into integer sequence by using the concept of Indexed based chaotic sequence [8]. In this approach a 1D logistic sequence is generated using (1) and then the indices of the elements in the original sequence is used as a separate sequence. This new sequence now consists of integer values.

5. PROPOSED MODEL

The proposed model consists two phases – an embedding phase and a decoding phase.

The embedding phase consists of two blocks namely the scene change detection block and the embedding block. The scene change detection block checks each frame for abrupt scene change using the histogram difference technique. It also divides the message into number of blocks depending upon the embedding capacity of a single frame in the video sequence.

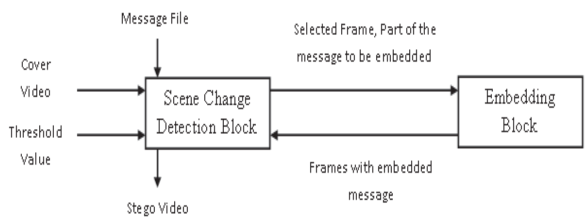

Fig.1. Block diagram of embedding phase.

When a frame is checked and confirmed for abrupt scene change, it is sent along with the block of message to be embedded to the embedding block. The embedding block embeds the message block in the frame using 3–3–2 approach of Least Significant Bit (LSB) substitution technique. In this approach the message character is converted into its 8 – bits bit stream. A pixel is then taken from the same position in the red plane, green plane and the blue plane of the frame. Then 3 LSBs of the pixel from the red plane is replaced with the first 3 bits of the message character bit stream, 3 LSBs of the pixel from the green plane is replaced with the next 3 bits of the character bit stream and 2 LSBs of the pixel from the blue plane is replaced with the last 2 bits of the message character bit stream. To signify the end of the message block a NULL character is appended to the message block as a delimiter. Also the pixel position are randomized using the integer sequence generated using the Indexed based chaotic sequence [8] to avoid embedding the message bits in the pixels sequentially. After the entire message has been embedded the ids of the frame used in embedding are embedded in the first frame of the video sequence.

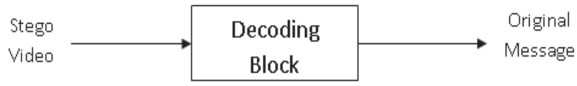

Fig. 1. Block diagram of decoding phase.

The decoding phase consists of a decoding block. The decoding block first extracts all the frame ids which are embedded in the first frame of the video sequence and then uses these ids to extract the message bits from the frames. These message bits are joined, converted into their character format and joined together to form the original message after eliminating the NULL character in the different message block. The algorithms designed for this model are as follows.

5.1 Algorithm for scene change detection

The following algorithm detects the frames in the cover video where the scene change has taken place and calls the embed function for each such frame until the entire message has been embedded. The algorithm is as follows.

1. Input the cover video v having
   $F \leftarrow$ Total number of frames
   $rows \leftarrow$ Height of the frame
   $cols \leftarrow$ Width of the frame
2. Break message m into message blocks m' [low_limit to up_limit] depending upon the amount of data to be embedded into a single frame.
3. $x \leftarrow rows/8$, $y \leftarrow cols/8$
4. for $i \leftarrow 1$ to $(F - 1)$ do
   a. $f1 \leftarrow$ frame(i), $f2 \leftarrow$ frame($i + 1$)
   b. for $j \leftarrow 0 : 8 : x - 1$ do
      for $k \leftarrow 0 : 8 : y - 1$ do
      i. block1$\leftarrow f1(j*8+1{:}j*8+8, k*8+1{:}k*8+8)$
      ii. block2$\leftarrow f2(j*8+1{:}j*8+8, k*8+1{:}k*8+8)$
      iii. $h1 \leftarrow$ histogram of block1, $h2 \leftarrow$ histogram of block2
      iv. $b \leftarrow \sum_{z=0 \text{ to } 255} |h1(z) - h2(z)|$
   c. Find the mean of values of b for all the blocks in the frames.
   d. If (mean > threshold AND up_limit <= length of message) then do
      i. Store $(i + 1)$ in $frame\_id$ array.
      ii. Call embed (frame (i + 1), m' [low_limit to up_limit], 8)
      iii. Replace the original frame with the modified frame.
5. Call embed (frame (1), $frame\_id$, 24)

5.2 Algorithm for embedding the message: embed(frame, array, number of bits b)

The following algorithm is use to embed the message. The embed function is called every time a frame is detected in which scene change has taken place. The embed function is also called a last time to embed the array containing the ids of the frames in which the scene change is detected and this array is embedded in the first frame because this frame is not used in the process of embedding the actual message. The algorithm is as follows.

1. Generate an integer sequence using Indexed based logistic sequence [8] using (1) with x_0 and μ as the initial value and store the sequence in array 'a'.
2. array $\leftarrow$ array + NULL character
3. Extract the red, green and blue plains of the frame
4. for $i \leftarrow 1$ to length of array
   a. Select a red, green and blue pixel at position $a(i)$.
   b. Select the message character at the i^{th} position and convert in b number of bits.
   c. Replace the bits in the LSB plane of the pixels in the red, green and blue plane with the bits of the elements in the array.
5. Regroup the modified red, green and blue planes of the frame and send it back to the scene change detection block.

5.3 Algorithm for decoding the stego video

The following algorithm is for decoding the stego video for extracting the original message. First the frame ids are extracted from the first frame and then the message is extracted from the frames identified by the frame ids. The algorithm is as follows.

1. Input stego video 'v' having
   $F \leftarrow$ total number of frames in the video.
   $rows \leftarrow$ height of the video.
   $cols \leftarrow$ width of the video.
2. Generate an integer sequence using Indexed based logistic map [] using same values as were used in

the embedding process and store the sequence in array '*a*'.

3. *f* ← frame (1)
4. Extract the red, green and blue plains of the frame *f*.
5. for *i* = 1 to length of *temp* do
   a. *id* ← *a* (*i*)
   b. Take the pixel at the position *id* from red, green and blue plains of *f*.
   c. Extract 4 LSBs each from red plain pixel, green plain pixel blue plain pixel of the frame *f*.
   d. Join the 12 bits and find its integer value *x*.
   e. *frameid* (*i*) ← *x* until NULL character is encountered.
6. for *i* = 1 to length of *frameid*
   a. *frm_id* ← *frameid* (*i*)
   b. *f1* ← frame (*frm_id*)
   c. *for j* = 1 to length of *f1*
      i. *id* ← *a* (*i*)
      ii. Take the pixel at the position *id* from red, green and blue plains of *f1*.
      iii. Extract 3 LSBs from red plain pixel, 3 LSBs from green plain pixel and 2 LSBs from blue plain pixel of the frame *f1*.
      iv. Join the 8 bits and find its character equivalent *ch*.
      v. *temp_msg* ← *temp_msg* + *ch* until NULL character is encountered.
   d. *original_message* ← *original_message* + *temp_msg*

6. DATA CAPACITY

The total amount of data that a single video frame can hold represents the embedding capacity of the frame.

Let *m* ← number of rows in the frame (height of the frame).

n ← number of columns in the frame (width of the frame).

Therefore the size of each frame is $m * n$.

Each frame in a color video is made of three components i.e. red, green and blue.

Thus the total number of pixel in a single frame is $m * n * 3$

In this model 8 bits of each message character are embedded in 3:3:2 format i.e. first 3 bits in the 3 LSBs of the pixel in the red component, next 3 bits in the 3 LSBs of the pixel in the green component and last 2 bits in the 2 LSBs of the pixel in the blue component. So to embed 8 bits we require 24 bits i.e. three 8 bits pixel. So the embedding capacity of the proposed model is $m * n$.

7. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

The proposed algorithm has been tested against three video files - cola.wmv, xylo.mpg and shuttle.avi as the cover video and 'ccs.txt' as the secrete message file. The output video 'colaout.avi', 'xyloout.avi' and shuttleout.avi' generated are the stego video of their respective cover videos.

The figures below show the frames of the cover video and the frames of the stego video in which the message has been embedded. The output shows absolutely no perceptual differences between the cover video and the stego video.

Fig. 3. Frame number 28 of the cover video 'cola.wmv' and stego video 'colaout.avi'.

Fig. 4. Frame number 99 of the cover video 'cola.wmv' and stego video 'colaout.avi'.

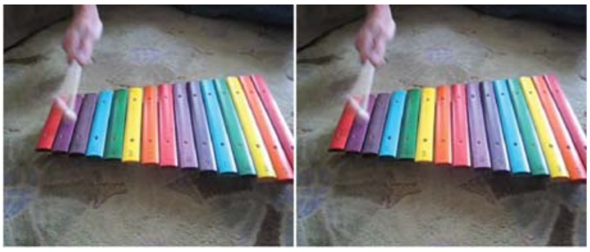

Fig. 5. Frame number 11 of the cover video 'xylo.mpg' and stego video 'xyloout.avi'.

Fig. 6. Frame number 12 of the cover video 'xylo.mpg' and stego video 'xyloout.avi'.

Fig. 7. Frame number 81 of the cover video 'shuttle.avi' and stego video 'shuttleout.avi'.

Fig. 8. Frame number 84 of the cover video 'shuttle.avi' and stego video 'shuttleout.avi'.

The perceptual quality of the cover video frame and the stego video frame has been evaluated using two quality evaluation metrics – Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR).

7.1. Mean Squared Error (MSE)

Mean Squared Error (MSE) is used to determine how much different the cover video frame and the stego video frame are. This is done by taking sum of difference between the corresponding pixel values of both the frames and then dividing the sum by the size of the frame. Since MSE determines the degree of dissimilarities between cover frame and stego frame therefore lower the value of MSE better is the result.

$$MSE = (\sum_{M,\,N} [f(m,n) - F(m,n)]^2) / M * N \qquad (2)$$

Where M and N are the rows and columns of the video frame, $f(m,n)$ is the pixel value at position (m,n) in the cover video frame and $F(m,n)$ is the pixel value at the position (m,n) in the corresponding stego video frame.

7.2 Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio (PSNR) is used to determine how much similar the cover video frame and the corresponding stego video frame are. Since PSNR determines the degree of similarity between the cover frame and the stego frame therefore higher the value of PSNR better is the result.

$$PSNR = 10. \operatorname{Log}_{10} (R^2 / MSE) \qquad (3)$$

Where R is the maximum possible value of luminance. For an 8 – bit image value of R will be 255. PSNR is measured in decibels (dB).

The MSE and PSNR values for the cover video frame and the stego video frame are shown in the following table

Table 3. Result of quality evaluation of cover video frames and stego video frames.

Cover Video	Stego Video	Frame Number	MSE	PSNR (in dB)
cola.wmv	colaout.avi	28	1.92	141.65
		99	0.70	146.04
xylo.mpg	xyloout.avi	11	2.00	141.47
		12	1.97	141.53
shuttle.avi	shuttleout.avi	81	2.03	141.41
		84	0.08	155.29

7.3 Histogram diffrence

Histogram of a video frame shows the distribution of intensities in the video frame. The histogram plots the frequency against the bins. The bin represents the pixel value that ranges from 0 to 255 and the frequency represents the number of times the pixel value appeared in the video frame. The histogram of the original video frame and its corresponding reconstructed video frame indicates the amount of difference between the two frames and helps to determine the quality of the reconstructed video.

The following figure shows the comparison between the histograms of the frames of the cover video and stego video.

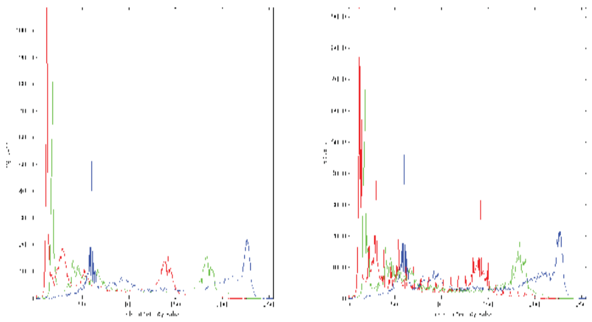

Fig. 9. Histogram comparison of frame number 28 of cover file cola .wmv and stego file colaout.avi.

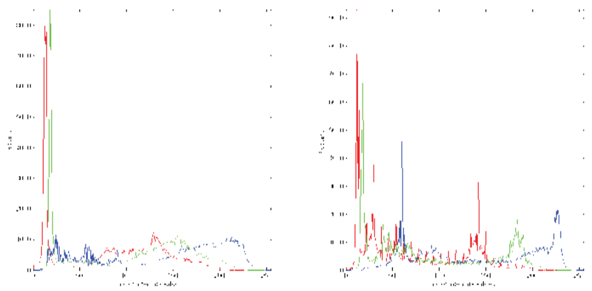

Fig. 10. Histogram comparison of frame number 99 of cover file cola .wmv and stego file colaout.avi.

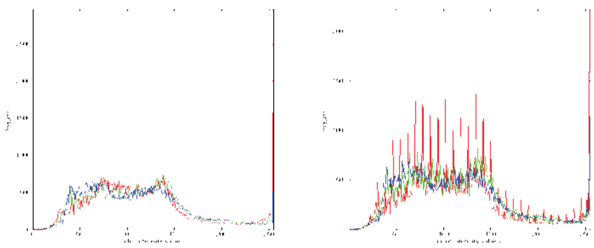

Fig. 11. Histogram comparison of frame number 11 of cover file xylo .mpg and stego file xyloout.avi.

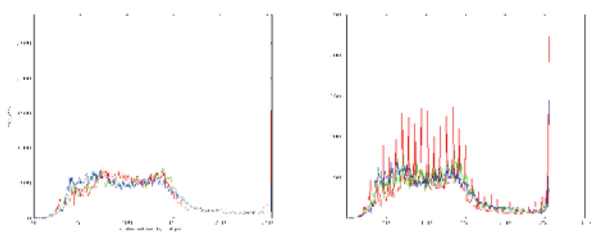

Fig. 12. Histogram comparison of frame number 12 of cover file xylo .mpg and stego file xyloout.avi.

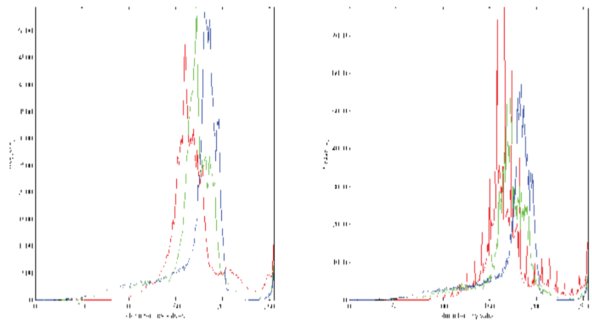

Fig. 12. Histogram comparison of frame number 81 of cover file shuttle.avi and stego file shuttleout.avi.

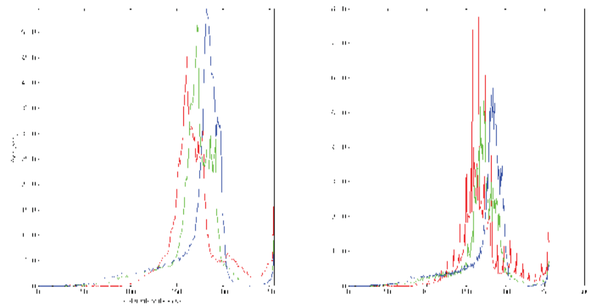

Fig. 13. Histogram comparison of frame number 84 of cover file shuttle.avi and stego file shuttleout.avi.

The comparison between the histograms of the frames of the cover video and the stego video shows some changes as a result of embedding the message in the frames. But the effect of these changes gets diluted in case of frames with abrupt scene changes. When the scene in a video sequence changes abruptly, histogram of the frame where the scene has changed abruptly also shows abrupt changes as compared to the histogram of the frames where the scene changes gradually. When the attacker will get hold of the stego video he will not have the original cover video to compare the histograms of the frame. As a result the changes caused in the histogram of the frame of the stego video after embedding the message will not reveal much information about something hidden inside that frame.

8. CONCLUSION

The proposed model thus takes the advantage of the abrupt scene change to hide the secrete message so that the attacker gets confused where the message is hidden in the video sequence. Also to increase the confusion the pixel position has been randomized using the integer sequence generated using the Indexed based chaotic sequence. The model has been tested across multiple video sequence and the results shows almost no perceptual difference between the original video and the reconstructed video i.e. the stego video with the secrete message.

9. REFERENCES

[1] Chen Ming, Zhang Ru, Niu Xinxin, and Yang Yixian, "Analysis of current steganography tools: Classifications & Features," Proceedings of the 2006 IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Dec 2006, pp. 384 – 387.

[2] Chi-Kwong Chan and L.M. Cheng, "Hiding data in images by simple LSB substitution," The Journal Of The Pattern Recognition Society, Published by Elsevier, pp. 469 - 474, 2003

[3] Dr. V. Vijayalakshmi, Dr. G. Zayaraz and V. Nagaraj, "A modulo based LSB steganography method," Proceedings of the 2009 IEEE International Conference On Control, Automation, Communication And Energy Conservation, 2009

[4] Kazem Ghazanfari, Shahrokh Ghaemmaghami and Saeed R. Khosravi, "LSB^{++} : An improvement to LSB^{+} steganography", Proceedings of IEEE TENCON 2011, 2011, pp. 364 – 368.

[5] Yi-Chun Liao, Chung-Han Chen, Timothy K.Shih and Nick C. Tang, "Data hidingin video using adaptive LSB," IEEE, 2009, pp. 185 – 189.

[6] Mohamed Elsadig, Miss Laiha Mat Kiah, Bilal Bahaa Zaidan and Aos Alaa Zaidan, "High rate video streaming steganography," Proceedings of the 2009 IEEE International Conference on Future Computer and Communication, 2009, pp. 672 – 675.

[7] Dolley Shukla and Manisha Sharma, "Scene change detection algorithms & techniques: A Survey," (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 12, pp. 73 – 77, December 2011.

[8] Aradhana Soni and Anuja Kumar Acharya, "A novel image encryption approach using an Index based chaos and DNA Encoding and its Performance Analysis," International Journal of Computer Applications, Volume 47– No.23, June 2012.