Video Steganography with LSB Color Detection

Arada Suttichaiya[1], Yuwarat Sombatkiripaiboon[1], Phet Imtongkhua[1], Chatchai Poonriboon[1], Chakchai So-In[1] and Paramate Horkaew[2]

[1]*Applied Network Technology (ANT), Department of Computer Science, Khon Kaen University, Khon Kaen, Thailand.*
[2]*School of Computer Engineering, Institute of Engineering Suranaree University of Technology, Nakhon Rachasima, Thailand.*
chakso@kku.ac.th

***Abstract*—Steganography is the method employed to prevent unsolicited access and malicious use of sensitive information. This research proposes an alternative approach to video steganography by exploiting Least Significant Bit (LSB) in the binary stream. Its main contribution is incorporating a color detection technique to optimize steganographic performance. The proposed encoding and decoding methods were implemented on MATLAB software to illustrate its applicability on uncompressed AVI movie. The results showed that it could conceal text, image, audio, and video data with non-secret streams. In this study, Peak Signal to Noise Ratio (PSNR) was used to assess its performance, whereby significant improvement over generic methods was found**

***Index Terms*—Color Detection; Head Frame; Least Significant Bit; Video Steganography.**

I. INTRODUCTION

Security in data communication and archiving is currently of utmost interest in the world of digital economy. Among the most valuable information needed protected are document and audiovisual contents involving both business transactions and personal assets. Steganography is therefore needed to conceal those data to prevent unsolicited access to those files without consent from communicating parties.

Without the necessary concealment, there are more than 556 million computer users worldwide (equivalent to 1.5 million a day or 18 people every second) currently being victimized [1]. These attacks can be categorized into computer virus, malware, worm, Trojan, and espionage both by the trusted insiders and general public. This type of computer crimes is considered the immediate threat to our society which affects both consumers and vendors alike and hence needs emergent remedy to prevent the unwarranted interceptions of such sensitive messages.

Nowadays, online media is a crucial means of broadcasting information to the public such as that managed by YouTube. According to the recent survey, during 2006 to 2013, the public interests in this kind of communication channels, e.g., contributors and viewers have increased by 72 percent [2].

Consequently, this research proposes a technique to conceal sensitive information with video stream. In this study, the least significant bits (LSB) in the pixel stream were replaced by those of hidden information. To this end, the detection of pixel colors and head frame designation was employed to determine a suitable ratio of concealment in each color component, so as to increase the amount of information being encoded.

This article is organized as follows. Section 2 gives background information of this research, and then related work will be briefly discussed in Section 3. Section 4 then explains our methodology for video steganography. Next, section 5 discusses the results obtained from performance evaluation. The conclusions and future work are also presented in the last section.

II. THEORY AND RELATED MATERIALS

This section explains the background theory and relating material for video steganography.

A. Image Processing

Image processing is a set of techniques which processes information in an image or a sequence thereof by means of a computer program. Its objectives are to extract quantitatively and qualitatively the required information, e.g., extent, shape, and motion direction of an object in the images. The extracted information would then by analyzed and used to build a system in various fields, such as fingerprint recognition, automatic postcode reader, and face identification.

B. Steganography

Steganography is a method for concealing secret data with general media, such as image, audio, video, and printed material. The method differs from cryptography in that the latter is probable to notice and if appropriate decrypting tool exists, it can be used to read the encrypted information, depending on various factors, such as the complexity of the code and algorithm. Steganography, on the contrary, hides that information behind sometimes mundane irrelevant data, ensconcing it from vicious intent. To tighten its security, the steganogram can be also encrypted with another method of choice.

C. Least Significant Bit (LSB) Steganography

LSB steganography is a technique which alters the least N significant bits of a pixel and replaces them with the message wish to be hidden [3]. For instance, a letter ‘A’ is to be hidden behind an 8 pixels image with 1 byte per pixel, whose binary values are shown below:

1001010**1**	0000110**1**	1100100**1**	1001011**0**
0000111**1**	1100101**1**	1001111**1**	0001000**0**

In this example, the bits with bold face are the least significant ones. Let the letter ‘A’ be represented by ASCII code, whose value is 01000001. Each bit in the code (underlined) will replace those marked above, resulting in another camouflage stream:

10010100	00001101	11001000	10010110
00001110	11001010	10011110	00010001

The merit of replacing the LSBs with those drawn from the message is that the resulted stream does not significantly differ from the original one, effectively hiding the message in a plain sight. Moreover, the amount of data (number of encoded bits) remains essentially the same.

D. Video

Video is a multimedia file which represents motion pictures with embedded sound. There are several types of video, e.g., educational video, entertainment video, and commercial video etc. A video file consists of three types of information as follows.

Image, whose dimensions characterized by its width and height in pixels unit. Audio, which can be characterized by its duration (second or sec), bit rate (kbps) and encoding format (such as mp3, wma, and wav). Video, which can be characterized by its frame rate as a speed of image sequence (fps), data rate as the number of bits that used to represent a motion picture in a unit of time (kbps), video sample size as the resolution each image in the sequence (bps), and video compression as a method used to encode the video into binary data.

E. RGB Color Representation

RGB is a basic computer color representation. It describes each picture element or pixel with its color components, i.e., red (R), green (G), and blue (B). A combination of these color components creates different true colors. The number of colors that can be rendered using this format depends on the precision of each component. Normally, a personal computer stores each component in an 8–bits memory, resulting in 256 possibilities for each basis color or 16 million combinations in total [4].

F. Audio Video Interleave (AVI) Video Format

AVI, standing for Audio Video Interleave, is a movie file that can be played simultaneously both motion pictures and audio through a computer. There are a number of computer programs that support this format, including Windows Media Player and Quick Time. AVI is considered a standard format for a computer running Windows operating systems. It contains very image and sound resolution, and therefore is normally large. There are variations of this format, i.e., DVD AVI is suitable for video editing as it best maintains the original video quality, while Xvid AVI is preferred in archiving as it is normally six times smaller than the original but with slight quality drop [4].

III. Literature Reviews

This section reviews the preliminary works that addressed video steganography as follows. In 2013, Yadav et al. [5] proposed a LSB steganographic method whose advantage was that the file size did not increase. Its main drawback, however, was that the locations that messages resided are systematic, and thus easily predictable. In the same year, Thakur and Saikia [6] used an 8 x 8 DCT of 128 x 128 which preprocessed a hidden image by using lossy compression.

Thanks to a unique DCT characteristic, the image corruption was minimized during the concealment, but it limits the scope of message types to only that of image. Later, Dasgupta et al. [7] applied a Genetic Algorithm (GA) and 3:2:2 LSB principles to steganography. In their study, GA was employed to examine the defects caused by the concealment in a video file. This resulted in minimal change between the original and post processed files Moreover, the noise occurred during the process was eliminated, resulting in less apparent camouflage.

Similar to others, their technique was still based on LSB method; and thus it was easily to predict the locations of hidden data. Bhole and Patel [8] also proposed an LSB steganography which randomly picked a set of data bytes bits at which messages were to replace. Compared to previous techniques, random locations made it harder to guess the place of hiding. Its limitation, however, is that it was suitable for text data only. Swathi and Jilani [9] similarly proposed an LSB based steganography whereby a polynomial expression was used to determine the alterable bit positions. To avoid the repetition should it occurred, 2 were added to resulted position, and the alterable bit were shifted accordingly.

Note that the advantage of this technique is that the added complexity due to polynomial calculation made the message harder to extract; but, it can conceal only text message. Involving higher level of image processing technique, Jain et al. [10] applied zero-crossing operator to determine the locations where edges passed on or close to. Those locations were then chosen as message hiding site. Despite higher computational demand, this method maintains hiding location regardless of image size.

IV. Methodology

The abovementioned techniques, based on LSB, DCT, and zero crossing has laid a common foundation on which our work was built. In this research, a video steganography based on LSB color detection is proposed. Specifically, color values were first examined in order to determine appropriate LSB format. Head frame where information was to be hidden was also specified.

With any LSB format (ratio) chosen, a fixed number of eight LSB bits per each single RGB pixel were allocated for concealment. These eight locations differed, depending on relative proportion of the RGB components, so as to ensure seamless message-image fusion, and so higher PSNR. This section, thus, discussed the proposed method in greater detail.

A. System Overview

Figure 1 illustrates the overview of the proposed technique. In this diagram, the processes of encoding and decoding the video steganography are depicted. During encoding process, an uncompressed AVI video was used as input and its size, and the number of frames were calculated. A head frame where the concealment occurred was also specified. Subsequently, the message to be hidden was converted into binary representation, whose size was computed to determine concealment capacity.

Note that each color component (Red, Green, and Blue) was averaged per frame to determine the appropriate LSB format, specifying the amount of LSB bits in each component replaceable by binary data. During the decoding process, the head frame was identified, and the steganogram pattern was then determined. Given the same LSB positions, the concealed data were extracted and arranged into the original message. The detailed implementation of encoding and decoding processes is provided the next sections.

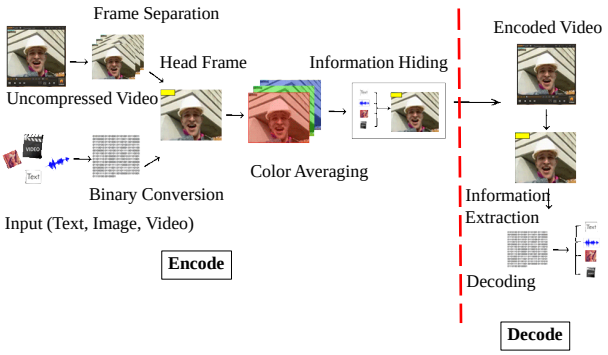

Figure 1: Overview diagram showing encoding and decoding processes

B. Encoding Process

This process consists of 7 steps as depicted in Figure 2. Here, each step in the diagram is described as follows. Firstly, a non-compressed video file in .AVI format was inputted into the system. Then, video size and number of video frames were computed; and the frame capacity that can be allocated for concealment was given by (WIDTH×HEIGHT×3)/8. The messages that could be concealed by the proposed method are text in .TXT format, image in .JPG or .JPEG format, audio in .WAV format, and also video in .AVI format. Before performing steganography, the message was converted into a binary (.BIN) stream, without header or structure, etc., by using dec2bin command, as illustrated in Figure 3.

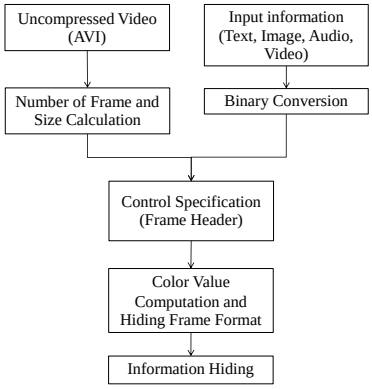

Figure 2: Diagram of the encoding process

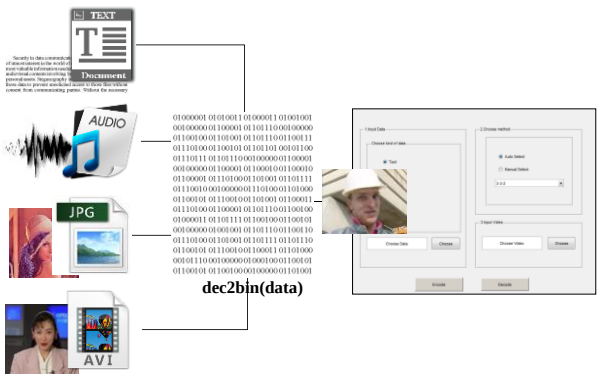

Figure 3: Acceptable message format and message to binary conversion

Note that the head frame identified in the previous step was used to store relevant meta data, i.e., number of frames, amount of concealed data, and concealment format/ pattern. The subsequent frames were then used to conceal the actual binary stream and the pattern/ format with respect to each frame. The number of frames required was computed by (Length of Input Data + Length of head frame)/ Frame Size, as shown in Figure 4.

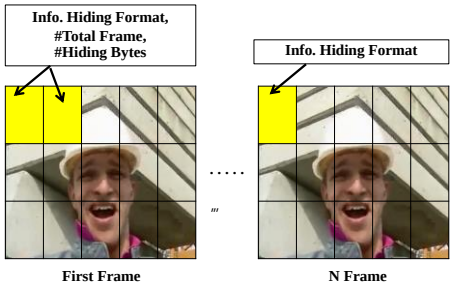

Figure 4: The layout of meta and concealed data in video frames

The next step was to compute the averaged values of color components (R, G, and B) in each frame, as shown in Figure 5. The steganography was then performed on the least 1, 2, 3, or 4 bits of each R, G, or B component, respectively, depending to the selected format (ratio). Specifically, 8 bits from the binary stream were distributed to R, G, and B components according to the format of that particular frame. Finally, all the concealing frames were put together into the original video sequence.

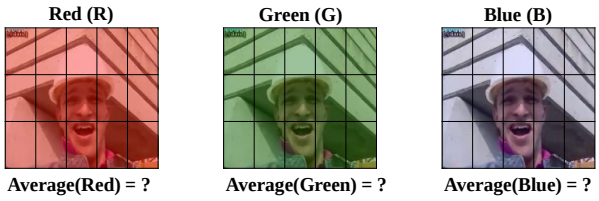

Figure 5: Computing averaged values of all color components

C. Decoding Process

This process, decoding, consists of three steps as follows. Similar to the encoding, it first took an uncompressed AVI video that has the message concealed. The next step was to extract the head frame and corresponding meta data to determine the number of frames, the amount of concealed data, and LSB format in the steganogram. The next subsequent N frames which contained the actual message would then be processed accordingly. Once the binary data were successfully drawn in respective pattern, they were then rearranged into their original stream, producing the hidden message (in text, image, audio, or video formats).

D. Color Detection

On examining the colors in each frame, their averaged values of red, green, and blue components were computed to determine appropriate concealing pattern in each pixel. Table 1 shows PSNR values of red, green, and blue colors, when concealing 1 – 4 bits data (R1 to R4; G1 to G4; and B1 to B4), respectively.

From Table 1, from the intensive evaluation, colors which had PSNR > 37 would be chosen, and their values fell between 64 and 255. Thus, the LSB formats were specified according to color intensities as follow.

Color Intensity > 192	=	High (H)
64 > Color Intensity > 192	=	Medium (M)
Color Intensity < 64	=	Low (L)

The specified range would then be used to decide suitable

LSB format by refereeing to Table 2. In each format, 8–bits data will be concealed within 1 pixel, consisting of R, G, and B components. Except that when all the components have high or low intensity, the medium and low concealing formats were chosen, respectively.

Table 1
PSNR of R/G/B color when concealing 1-4 bits data

R1/G1/B1	R2/G2/B2	R3/G3/B3	R4/G4/B4
64	128	129	255
94.47, N/A, 94.47	94.47, 101.25, 94.47	102.48, 109.30, 102.48	N/A, 114.87, N/A
67.04, 78.98, 67.04	80.91, 86.51, 80/31	94.47, 88.21, 88.82	105.63, 98.78, 105.63
57.32, 60.40, 54.93	70.17, 72.83, 69.97	75.36, 77.93, 75.20	85.89, 82.89, 85.89
43.13, 45.54, 43.13	54.93, 56.08, 54.89	62.28, 62.24, 62.28	68.90, 67.60, 68.95

Table 2
Concealing LSB format from color detection

Red	Green	Blue	Format
High	High	Low	3-3-2
High	Low	High	3-2-3
Low	High	High	2-3-3
High	Low	Low	4-2-2
Low	High	Low	2-4-2
Low	Low	High	2-2-4
High	Medium	Low	4-3-1
High	Low	Medium	4-1-3
Medium	High	Low	3-4-1
Medium	Low	High	3-1-4
Low	Medium	High	1-3-4
Low	High	Medium	1-4-3
High	High	High	3-3-3
Low	Low	Low	1-1-1

For example, suppose the averaged red is 240.5 which is greater than 192, and thus was put in High range. Following this criteria, green and blue whose averages are 63.1 and 190 which are lower than 64 and between 64 and 192, respectively. The blue and green colors are then put in Low and Medium ranges, respectively. For this particular image, the suitable format is thus 4-1-3 (High – Low – Medium), according to Table 2.

E. Head Frame

Head frame is the first part in a frame that was allocated for storing relevant meta data and hidden message. In the first frame, this space was used to store number of steganographic frames and the amount of data, which will be hidden in the subsequent frames. More specifically, in the first head frame, the (1, 1) pixel is reserved for the LSB format. The next 8 bits are reserved for number of frames involved, and another 14 bits later are used for the amount of data hidden. In the subsequent frames, their (1, 1) pixels store only the LSB format used in each particular frame.

V. PERFORMANCE EVALUATION

This section explains the strategies adopted in evaluating the performance of the proposed technique.

A. Design of Experiments

In our experiments, five types of videos were used to evaluate the proposed technique. They are stated as follows [11]:

1. News had a size of 352 x 288 pixels and consisted of 301 frames and its size of this 10 seconds video are 80 MB.
2. Moving Person was 352 x 288 pixels in size, consisting of 295 frames, and its size of this 9 seconds video are 85.6 MB.
3. Sports had a size of 800 x 600 pixels and consisted of 230 frames, and its size of this 7 seconds video are 315 MB.
4. Landscape was 1280 x 720 pixels in size, consisting of 137 frames, and its size of this 5 seconds video are 316 MB.
5. Animation had a size of 1024 x 575 pixels and consisted of 129 frames, and its size of this 4 seconds video are 217 MB.

In this setup, four types of messages were employed as the hidden data. Each type had 3 different files in the same format, i.e.,

1. Text consists of Text1.TXT, Text2.TXT and Text3.TXT whose sizes are 107 KB, 214 KB, and 429 KB, respectively.
2. Picture consists of Pic1.JPG, Pic2.JPG and Pic3.JPG whose dimensions are 512 x 512, 1160 x 870, and 2048 x 1536 pixels and whose sizes are 89.6 KB, 179 KB, and 1039 KB, respectively.
3. Voice and Audio consists Audio1.WAV, Audio2.WAV, and Audio3.WAV whose lengths are 2, 3, and 4 seconds and whose sizes are 55 KB, 534 KB, and 740 KB, respectively.
4. Moving Picture consists of Video1.AVI, Video2.AVI, and Video3.AVI whose dimensions and durations are 40 x 22 pixels 6 seconds, 70 x 38 pixels 4 seconds, and 90 x 49 pixels 2 seconds; and whose sizes are 595 KB, 727 KB, and 595 KB, respectively.

In the following evaluations, five different types of videos were examined, each when concealing four types of messages, i.e., text, image, audio, and video. The steganographic streams obtained from the proposed LSB technique based on color detection were compared against generic LSB techniques with steganographic ratio of RGB components of 1:1:1 and 4:4:4, respectively. Specifically, their performances were assessed by comparing the resulted streams with their respective originals and by comparing the number of frames used for concealing.

To this end, four main experimental cases were devised. The first one was performing steganography on three different .TXT text files, using five video types (News, Moving Person, Sports, Landscape, and Animations). The second, third, and fourth cases were also carried out on the identical set of videos but doing so on .JPG images, .WAV audios, and .AVI videos, each of which used three different files in their respective formats.

B. Measurements

In experiments described here, Peak Signal to Noise Ratio (PSNR) [12] was employed to assess the performance of the techniques. Herein, the PSNR was defined as a ratio between the peak of resulted video stream (Signal) and the difference between this stream and its original (Noise). The PSNR was thus expressed as the equation below.

$$PSNR = 10log_{10}\frac{Max^2}{MSE} \quad (1)$$

In this equation, Max is the maximum pixel intensity of a frame and MSE (Mean Squared Error) is defined as the mean of squared difference between resulted and original frames, i.e.,

$$MSE = \frac{1}{(m\,n)}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (2)$$

Note that here, m and n is the size of the frames, I and K are the resulted and original frames, and (i, j) is the pixel coordinates.

It is worth noting that observing that the acceptable results are defined as the steganogram with the PSNR not lower than 37.

C. Experimental Results

The evaluations reported herein followed these steps. The encoding process started by first (1) selecting the type of message to be hidden (Text, Image, Audio, or Video) and then browsing the required file (2). The user was then asked to decide which technique to be applied between auto and manual selection (3).

When choosing auto mode, the LSB format determined by the proposed color detection would be used, while manual mode allowed user to set the LSB format (Red: Green: Blue) (4). Next, the source videos were chosen from 5 available uncompressed AVI files (5). Encoding command finally started the steganography using the provided parameters and data (6). Figure 6 shows the GUI of this process.

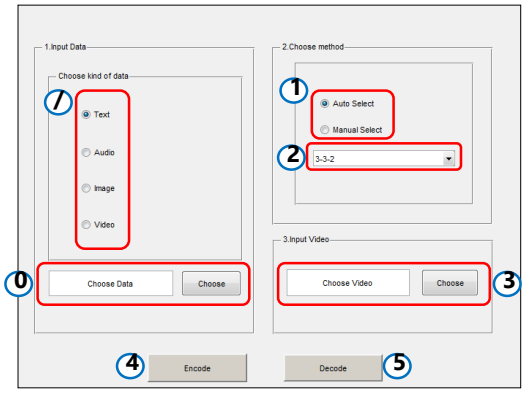

Figure 6: Graphic user interface (GUI) of encoding and decoding

On decoding a message from a video, first its type needed to be specified (1). Then, the encoded stream was chosen and Decode command was executed, i.e., (5) and (7), respectively.

Note that the encoded stream in each message-video combination as described in Section 4 Methodology was compared with their original. Table 3 shows the PSNR and number of frames used in the concealing text message in the videos.

It can be concluded from the Table 3 comparing the LSB video steganography on text messages with 1:1:1, 4:4:4, and automatically selected format by color detection that, the 1:1:1 method gave the highest PSNR, followed by color detection and 4:4:4 LSB methods, respectively. Comparing the number of frames used revealed that the 4:4:4 LSB method required the least number of frames to conceal such messages, followed by color detection and 1:1:1 LSB methods.

Table 3
PSNR/Number of frames used for text steganography

Video	Size (Bytes)	LSB (1:1:1)	LSB (4:4:4)	LSB (CD)
Akiyo	110,000	69.87/4	56.28/1	60.47/2
Akiyo	220,000	67.45/7	53.26/2	58.70/3
Akiyo	440,000	67.77/13	50.19/4	56.44/5
Foreman	110,000	70.58/3	56.18/1	60.46/2
Foreman	220,000	67.82/6	53.18/2	58.73/3
Foreman	440,000	64.93/12	51.73/3	64.93/5
Aspen	110,000	68.31/1	48.26/1	55.98/1
Aspen	220,000	68.32/1	48.37/1	55.97/1
Aspen	440,000	65.43/2	48.50/1	56.00/1
Minion	110,000	72.29/1	52.13/1	59.82/1
Minion	220,000	72.22/1	52.24/1	59.78/1
Minion	440,000	69.20/2	52.31/1	59.74/1
Sport	110,000	74.81/1	54.46/1	61.52/1
Sport	220,000	71.64/2	54.44/1	61.53/1
Sport	440,000	69.00/3	54.60/1	61.67/1

The same conclusions can be drawn from Table 3 in the image case, as shown in Table 4, which are that the 1:1:1 LSB method produced the highest PSNR, and that the 4:4:4 LSB method needed the least number of frames. To further emphasize these findings, the steganography was also performed on audio and video messages. Their PSNR and frame counts were shown in Tables 5 and 6, respectively.

Table 4
PSNR/Number of frames used for image steganography

Video	Size (Bytes)	LSB (1:1:1)	LSB (4:4:4)	LSB (CD)
Akiyo	110,000	71.14/3	56.13/1	63.86/1
Akiyo	220,000	68.13/6	53.12/2	60.89/2
Akiyo	440,000	61.01/31	47.45/8	53.06/12
Foreman	110,000	70.58/3	56.02/1	63.87/1
Foreman	220,000	68.55/5	52.92/2	60.87/2
Foreman	440,000	61.32/28	48.13/7	53.48/11
Aspen	110,000	68.30/1	48.24/1	56.04/1
Aspen	220,000	68.31/1	48.34/1	56.11/1
Aspen	440,000	62.43/4	49.00/1	53.30/1
Minion	110,000	72.33/1	52.19/1	59.89/1
Minion	220,000	72.31/1	52.26/1	59.90/1
Minion	440,000	65.28/5	49.44/2	57.16/3
Sport	110,000	75.04/1	54.91/1	62.00/1
Sport	220,000	72.17/2	54.80/1	62.13/1
Sport	440,000	67.00/6	51.84/2	57.35/3

Table 5
PSNR/Number of frames for audio steganography

Video	Size (Bytes)	LSB (1:1:1)	LSB (4:4:4)	LSB (CD)
Akiyo	110,000	72.89/2	55.76/1	63.65/1
Akiyo	220,000	63.88/16	50.00/4	55.86/6
Akiyo	440,000	62.50/22	48.39/6	54.77/9
Foreman	110,000	72.12/2	55.52/1	63.53/1
Foreman	220,000	63.98/15	49.96/4	55.89/6
Foreman	440,000	62.75/20	49.28/5	54.77/8
Aspen	110,000	68.28/1	48.19/1	55.99/1
Aspen	220,000	65.41/2	48.52/1	56.23/1
Aspen	440,000	63.67/3	48.68/1	56.34/1
Minion	110,000	72.35/1	52.16/1	59.84/1
Minion	220,000	67.52/3	52.46/1	60.10/2
Minion	440,000	66.25/4	52.51/1	57.00/1
Sport	110,000	75.09/1	55.26/1	62.20/1
Sport	220,000	69.06/4	55.49/1	59.42/2
Sport	440,000	67.82/5	51.96/2	59.11/2

Table 6
PSNR/Number of frames for video steganography

Video	Size (Bytes)	LSB (1:1:1)	LSB (4:4:4)	LSB (CD)
Akiyo	110,000	63.37/18	48.54/5	54.86/7
Akiyo	220,000	62.50/22	48.04/6	53.95/9
Akiyo	440,000	63.37/18	48.72/5	55.01/7
Foreman	110,000	63.45/17	48.42/5	54.90/7
Foreman	220,000	62.75/20	48.85/5	54.51/8
Foreman	440,000	63.70/16	49.72/4	55.73/6
Aspen	110,000	65.40/2	48.30/1	55.99/1
Aspen	220,000	63.66/3	48.49/1	56.14/1
Aspen	440,000	65.40/2	48.39/1	56.10/1
Minion	110,000	67.54/3	52.26/1	56.80/2
Minion	220,000	66.27/4	52.49/1	56.89/2
Minion	440,000	67.53/3	52.39/1	56.89/2
Sport	110,000	69.02/4	56.26/1	59.46/2
Sport	220,000	67.99/5	52.78/2	59.60/2
Sport	440,000	68.98/4	56.51/1	59.54/2

It is evident that in all cases the PSNR is acceptably high. Although the 1:1:1 LSB offered highest PSNR, it required the most number of frames to conceal the messages, compared to 4:4:4 LSB and color detection methods. On the contrary, the 4:4:4 LSB required the least number of frames, at the cost of scarifying PSNR in the resulted streams. It is therefore safe to note here that, LSB format automatically determined by color detection offered the optimal compromise between PSNR and number of frames required to conceal the messages.

To elucidate the above observation, the graphs comparing PSNR of resulted steganography using the three LSB methods on different types of message in each video are shown in Figure 7 (a to e). It can be clearly seen in these graphs that 1:1:1 LSB gave consistently high PSNR values, regardless of message types. The LSB by using color detection offered slightly lower PSNR but still higher than the 4:4:4 LSB. It is worth noted that the differences are small when concealing video messages.

VI. Conclusion and Future Work

This research proposed the use of color detection in the LSB bits (LSB Color Detection) to adjust ratios of concealing data according to respective pixel RGB values.

With this technique the number of bits replaced by steganogram increased from that was possible in the conventional 1:1:1 method, effectively improving the steganographic performance. In addition, when compared with the 4:4:4 LSB method, the proposed LSB color detection generated video steganography with higher PSNR.

When considering the number of frames needed to encode these messages, it was found that color detection based LSB can effectively conceal a similar amount of data compared to 4:4:4 LSB, given a specific number of frames. However, unlike 4:4:4 LSB, it could do so without having to scarify the PSNR.

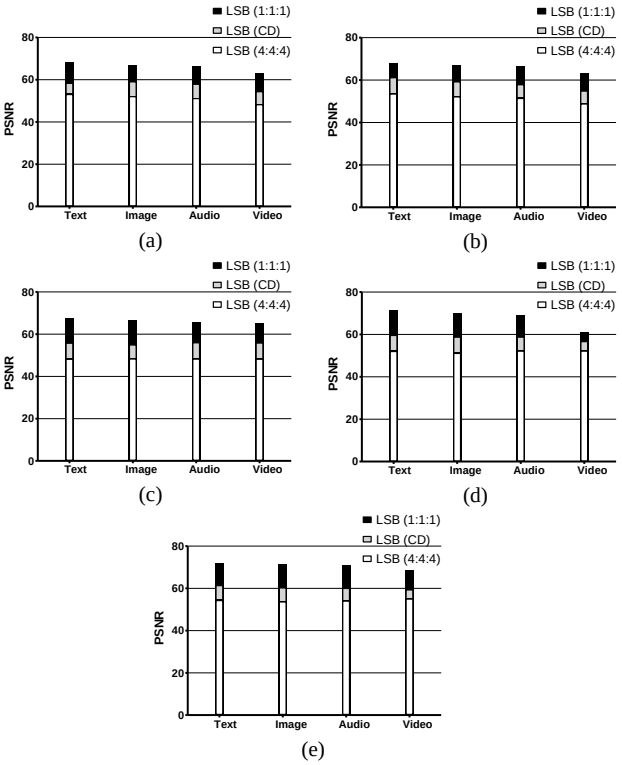

Figure 7: Encoding performance (PSNR) with different file formats: (a) Akiyo, (b) Foreman, (c) Aspen, (d) Minion, and (e) Sport

References

[1] it24hrs. 2014. The shift in hacker attack behavior so as to gain higher reward. www.it24hrs.com/2014/data-breach-impact.

[2] Gesengues, A. 2013. Study: Number Of People Who Share Or Post Videos Online Has Doubled Since 2009, Pew Research Center.

[3] Pavani, M. Naganjaneyulu, S. and Nagaraju, C. 2013. A Survey on LSB Based Steganography Method. Int. J. Of Engr. and Comput. Sci. 2(8): 2464–2468.

[4] Hunt, R.W.G. 2004. The Reproduction of Colour. Wiley–IS&T Series in Imaging Sci. and Technol., 724 .

[5] Yadav, P. Mishra, N. and Sharma, S. 2013. A Secure Video Steganography with Encryption Based on LSB Technique," Proc. IEEE Int. Conf. on Comput. Intell. and Comput. Research. 1–5.

[6] Thakur, V. and Saikia, M. 2013. Hiding Secret Image in Video. Proc. Int. Conf. on Intell. Syst. and Signal Process. 150–153.

[7] Dasgupta, K. Mondal, J. K. and Dutta, P. 2012. Hash based Least Significant Bit Technique for Video Steganography (HLSB), Int. J. of Sec., Privacy and Trust Management. 1(2): 1–11.

[8] Bhole A. T. and Patel, R. 2012. Steganography over Video File using Random Byte Hiding and LSB Technique, Proc. IEEE Int. Conf. on Comput. Intell. and Comput. Research. 1–6.

[9] Swathi A. and Jilani, S.A.K. 2012. Video Steganography by LSB Substitution Using Different Polynomial Equations. Int. J. of Comput. Engr. Research. 2(5): 1620–1623.

[10] Jain, N. Meshram, S. and Dubey, S. 2012. Image Steganography Using LSB and Edge–Detection Technique. Int. J. of Soft Comput. and Engr. 2(3): 2231–2307.

[11] Video Dataset (Producer, Marin Kocs, Nbadleague). 2015. Akiyo, Foreman, and Aspen [media.xiph.org/video/derf]; Funny Video 3D and Nbadleague [www.youtube.com]

[12] Huynh-Thu, Q. and Ghanbari, M. 2008. Scope of validity of PSNR in image/video quality assessment. Electro. Letters. 44(13): 800–801.