PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

A Technical Mini Project Report on

'HIDING A SECRET IMAGE IN A VIDEO USING LSB SUBSTITUTION TECHNIQUE'

Submitted in Partial fulfillment of the Requirements for VI Semester

Bachelor of Engineering
In
Electronics and Communication

By

AJAY BHIMASHANKAR
(01FB15EEC016)

AMOGH M K
(01FB15EEC027)

ARVIND K
(01FB15EEC049)

Jan. - Apr. 2018

Under the guidance of
Mrs. Preethi S.J
Asst. prof, Dept. of ECE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PES UNIVERSITY, BANGALORE-85

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Technical Mini Project Report entitled

‘HIDING A SECRET IMAGE IN A VIDEO USING LSB SUBSTITUTION TECHNIQUE’

is a bonafide work carried out by

AJAY BHIMASHANKAR (01FB15EEC016)

AMOGH M K (01FB15EEC027)

ARVIND K (01FB15EEC049)

In partial fulfillment for the completion of VI semester “Digital Image Processing” Elective III course work in the Program of Study B.Tech in Electronics and Communication under rules and regulations of PES University, Bengaluru during the period Jan.2018 – Apr. 2018. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report.

Signature with date & Seal
Internal Guide

Signature with date & Seal
Chairperson

Name/s of the student/s **AJAY BHIMASHANKAR (01FB15EEC016)**
Name/s of the student/s **AMOGH M K (01FB15EEC027)**
Name/s of the student/s **ARVIND K (01FB15EEC049)**

Acknowledgement

On the submission of our project entitled **‘Hiding a secret image in a video using LSB substitution technique’**, we would like to express our gratitude to the project supervisor and advisor **Mrs. Preethi S.J.**, Asst. Professor, Dept. of ECE, for her guidance and inspiration during the course of the project work.

Also, we would like to thank our family and friends for their support and encouragement.

Abstract

Video Steganography is a technique to hide any kind of files into a carrying Video file. The use of the video based Steganography can be more eligible than other multimedia files, because of its size and memory requirements. The least significant bit (LSB) insertion is an important approach for embedding information in a carrier file. Least significant bit (LSB) insertion technique operates on LSB bit of the media file to hide the information bit. In this paper, we introduced a data hiding scheme to hide the information in specific frames of video and in specific location of the frame by LSB substitution using polynomial equation

Keywords - Steganography, Stego-Video, Cover frame, Embedding, Extracting, Secret image, LSB (least significant bit), Data Security.

Table of Contents

1. Introduction

1.1 Overview-Background

Steganography is a Greek word which means “covered or hidden writing”. The idea of steganography is thousands of years old. The Greek soldiers used to pass secret message they shave a slave’s head, tattoo a message on his head when the hair grow again then the tattoo could not be seen. Receiver shaves the head of slave and gets the message from the tattoo. Invisible ink was also used during the World War II. Watermarking and fingerprinting are closely related to steganography. Data hiding can be used for secret transmission. Steganography is a technique for hiding secret information in digital image, audio and video to secure information from third party Steganography apply in various fields such as military and industrial applications. Lossless steganography techniques are use for secure and successful transmission of information from sender to receiver. Usually, steganography was based on hiding secret message in digital image files. Recently, the computer programmers start interest applying steganographic techniques to video files as well as audio files.

1.2 Brief description

Video Steganography mechanism is used to hide data like secret images and any other files within a video. Today on the internet the most popular image formats are Graphics Interchange Format (GIF), Portable Network Graphics (PNG) and Joint Photographic Experts Group (JPEG). Most of the advanced techniques not use the structures of these formats but they use the Bitmap format (BMP) for its easy data structure .We use digital images for steganography because of the weaknesses in the HVS (human visual system) which has a low sensitivity in random pattern changes. Due to this weakness the secret image can be hiding into the cover video or image without being noticed. A digital video contains a set of frames (digital images) which are played back at fixed frame rates based on the video standards. An image is a collection of pixels and each pixel is a mixture of three primary colors RGB (Red, Green and Blue). Pixels in the image are show row by row horizontally. Data hiding in the video/image get less troubled as contrasted to other multimedia files. When data is hiding in an image its size increase .So compression techniques are required. There are two types of compression techniques lossy and lossless. Video/image size can be decreased by compression technique. There are various embedding techniques that enable us to hide secret message in a given object. Meanwhile, whole methods definitely assure almost all the requirements so that steganography can be apply accurately. Steganography techniques must satisfy these following requirements:

a) The integrity of the hidden data must be accurate after embedding it inside the stego object.

b) Robustness- The hidden data should be survived through any processing operation through which host signal undergoes and protect its loyalty.

c) Capacity-Maximize data embedding payload.

d) The stego object must stay unmodified or almost unmodified to the bare eye.

e) Security- Use a security key.

1.3 Problem Definition

To implement video Steganography in Matlab for hiding secret image in the cover frames of a carrier video using LSB (Least Significant Bit) modification technique and retrieving the hidden image from the video at the receiver end.

2. Literature survey

There are numerous techniques for hiding data in a digital container file. Although this project focuses solely on using a video cover file, there are techniques in audio and image steganography that still bear relevance to video file formats. Furthermore, video can be split into two components: the audio stream and the picture stream. To be able to work with video steganography, it is important that we understand the audio and image (picture) techniques that have already been developed and explored within digital steganography.

There are a variety of steganographic techniques that can be used to conceal information in a container file. The steganographic techniques discussed herein fall into one of the following categories which are defined by the method of data hiding*: injection, substitution, generation and transform domain.*

2.1 Injection Techniques

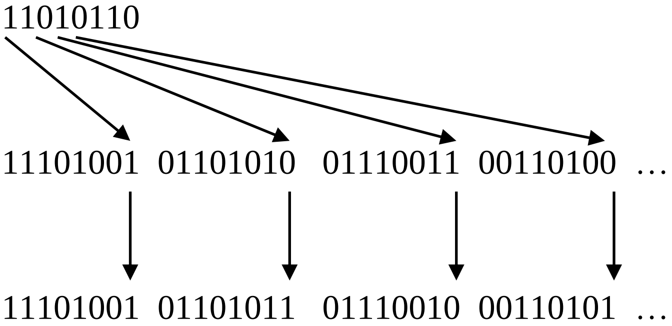
Steganography performed by injection is by far the simplest steganographic technique. As the name suggests, data is injected into redundant areas of the container file. Most files have an End of File (EOF) marker or a file size marker, which indicates where the reading of a file should cease. Data can be placed at the end of the file (after the EOF marker), without affecting the integrity of the container file. This technique is very simple and as such, is very easy to detect.

The nature of injection techniques means that it is a fairly straightforward process to detect and extract the covert data. Techniques that embed the data into the container (via generation or substitution) are generally harder to detect because the covert data is interwoven with the original container data, thus making it harder to identify the presence of covert data.

2.2 Substitution Techniques

A substitution technique will identify areas of a file of least relevance, and replace this data with the covert data. This technique does not modify the size of the container file, and is consequently limited by the steganographic capacity of the file.

Least Significant Bit Substitution Technique:-



One of the most common steganographic techniques is Least Significant Bit (LSB) manipulation. LSB manipulation can be easily applied to some audio and image formats, and works by modifying part of the representation of the data stored within the container format.

2.3 Generation Techniques

Generation techniques involve generating a container file based on the covert data that is to be embedded. With a generation technique there is no original container file because the cover object is completely generated, this provides a unique advantage over other steganographic techniques that required an existing input container.

But it has a potentially huge disadvantage. If the original (unmodified) container file exists, or is leaked, outside the secure domain this can provide an attacker with significant information that can accelerate a steganalysis attack. For instance, consider Alice and Bob generate car images (consider them as car enthusiasts so that exchanging pictures of them doesn’t raise suspicion) as a container file and exchange data by hiding it in these images. In this case if an attacker gets his hands on one of the unmodified container files (without hidden data), then he can use this file with its respective modified file (with hidden data) to find the technique in which they are hiding the data and hence access all the data previously exchanged between them.

Another disadvantage of this technique is that it should generate container files that fits the profile of those communicating which is a complex process and time consuming.

2.4 Transform Domain Techniques

Transform domain techniques are generally used on compressed container files. For instance, data hiding in JPEGs is commonly achieved by operating in the frequency domain and modifying the Discrete Cosine Transform (DCT). This technique is relatively basic, and numerous steganalysis methods have been developed which are easily capable of detecting covert data that is embedded using this method.

3. Software Requirements

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Matlab,stands for Matrix Laboratory, is a complete programming environment that encompasses its own programming language, IDE (integrated development environment), libraries (called toolboxes in Matlab).

4. Software Design

Algorithm:-

For Embedding

Step 1: Input video object file.

Step 2: Split the video into frames.

Step 3: Acquire the frames in which the secret image has to be hidden using the key.

Step 4: Split the secret image into 8 parts and store them in the LSBs of the acquired cover frames.

Step 5: Rejoin the cover frames with the rest of the frames and regenerate video.

For Extracting

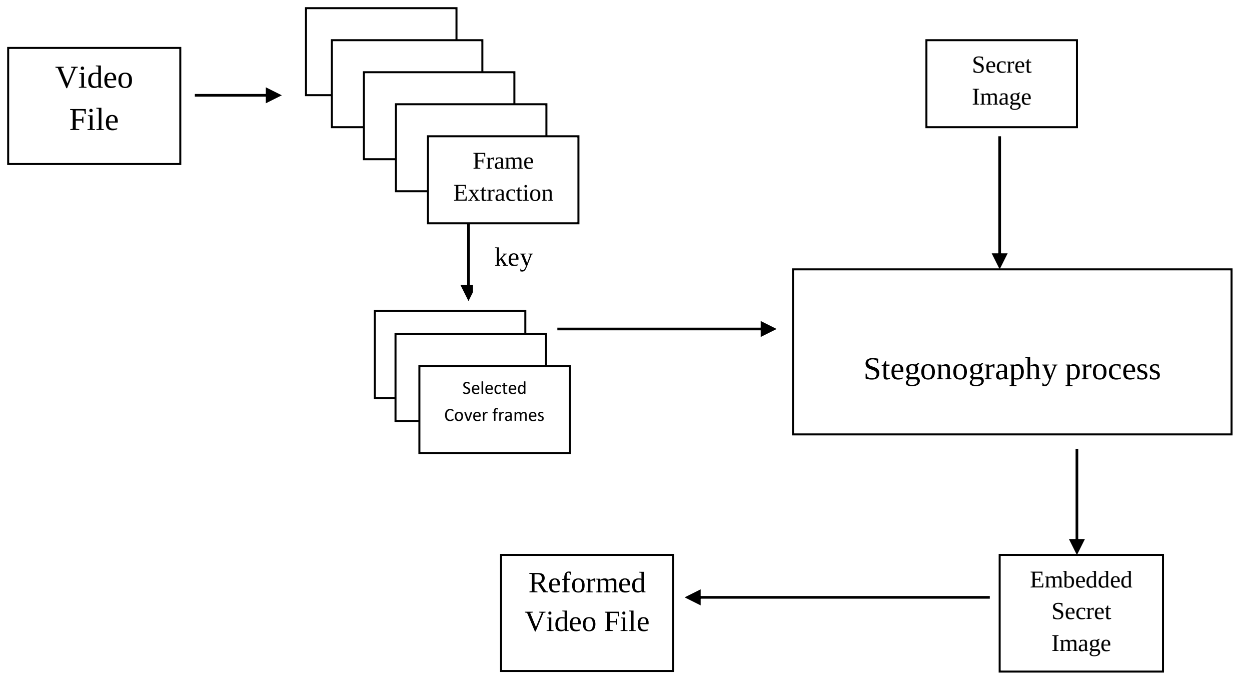
Step 1: Input stego video file.

Step 2: Split video into frames.

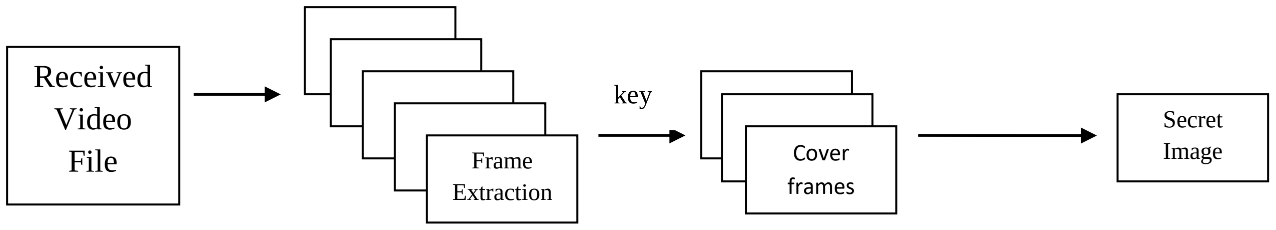
Step 3: Find and extract the cover frames using the key.

Step 4: Recover the hidden image by extracting the data stored in the LSBs of the cover frame.

Block Diagram for embedding



Block Diagram for extracting at receiver



5. Implementation

Step1:- Code for converting video to stream of images:

```
workingDir = 'E:\college\sem 6\DIP\Project';

mkdir(workingDir,'videoimages');

vid=VideoReader('video.mp4');

i=1;

while hasFrame(vid)

   img=readFrame(vid);

   filename=[sprintf('%03d',i) '.png'];

   fullname=fullfile(workingDir,'videoimages',filename);

   imwrite(img,fullname);

   i=i+1;

end
```

Step 2:- Code for hiding an image in cover frames selected by key:

```
key='1024';                    // Security Key used is 1024

a=key(1)-'0';

b=key(2)-'0';

c=key(3)-'0';

d=key(4)-'0';

imgtobehidden=imread('image.jpg');

imgtobehidden=double(imgtobehidden);

sizeofimage=size(imgtobehidden);

imgtobehidden=imgtobehidden(:);

imagenomat=[];


workingDir = 'E:\college\sem 6\DIP\Project';
```

```
mkdir(workingDir);
mkdir(workingDir,'videoimagesafterhiding');
for iter1=1:8
    imageno=a*(iter1^3)+b*(iter1^2)+c*(iter1)+d;
    imagenomat=[imagenomat imageno];
    imagename=[sprintf('%03d',imageno) '.png'];
    image=imread(imagename);
    image=image(:);
    for iter2=1:size(image)
        if rem(image(iter2),2)==1
            image(iter2)=image(iter2)-1;
        end
        image(iter2)=image(iter2)+rem(imgtobehidden(iter2),2);
        imgtobehidden(iter2)=floor(imgtobehidden(iter2)/2);
    end
    image=reshape(image,sizeofimage(1),sizeofimage(2),sizeofimage(3));
    fullname=fullfile(workingDir,'videoimagesafterhiding',imagename);
    imwrite(image,fullname);
end
%{
for iter3=1:1525
    if(any(imagenomat(:)==iter3))
        continue
    end
    imagename=[sprintf('%03d',iter3) '.png'];
    image=imread(imagename);
    fullname=fullfile(workingDir,'videoimagesafterhiding',imagename);
    imwrite(image,fullname);
end
%}
```

Step 3:-Code for reforming a video file (creation of stego-video):

```
workingDir = 'E:\college\sem 6\DIP\Project';
vid=VideoReader('video.mp4');
video=VideoWriter(fullfile(workingDir,'originalvideo'));
video.FrameRate=vid.FrameRate;
open(video);
for i=1:1525
    imagename=[sprintf('%03d',i) '.png'];
    img=imread(fullfile(workingDir,'videoimages',imagename));
    writeVideo(video,img)
end
```

Step 4:-Converting stego-video into stream of images at the receiver:

Same code as used in step 1.

Step 5:-Code for recovering image from cover frames:

```
a=key(1)-'0';
b= key='1024';
key(2)-'0';
c=key(3)-'0';
d=key(4)-'0';
sampleimg=imread('007.png');
sizeofimg=size(sampleimg);
recoveredimg=uint8(zeros(sizeofimg));
for iter1=1:8
```

```
    imageno=a*(iter1^3)+b*(iter1^2)+c*(iter1)+d;
    imagename=[sprintf('%03d',imageno) '.png'];
    image=imread(imagename);
    recoveredimg=recoveredimg+(2^(iter1-1))*rem(image,2);
end
imshow(recoveredimg);
imwrite(recoveredimg,'recovered image.png');
```

Results:-

Secret Image hidden in cover frames

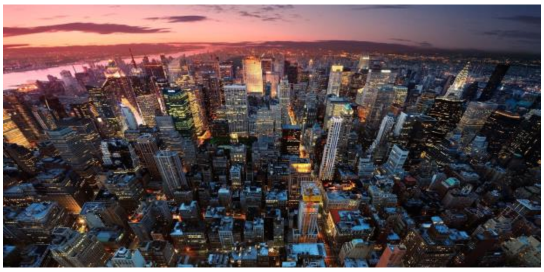
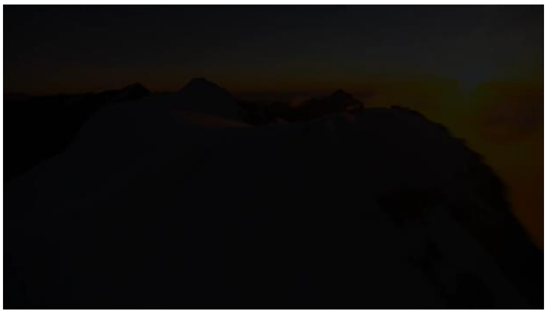


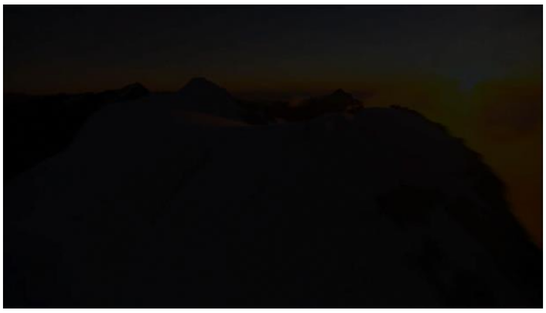
Image obtained upon extraction

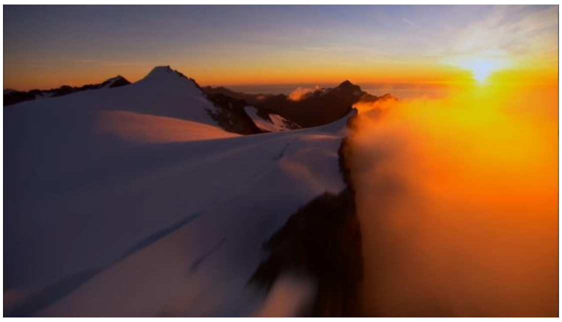
Cover frames before and after hiding secret image:

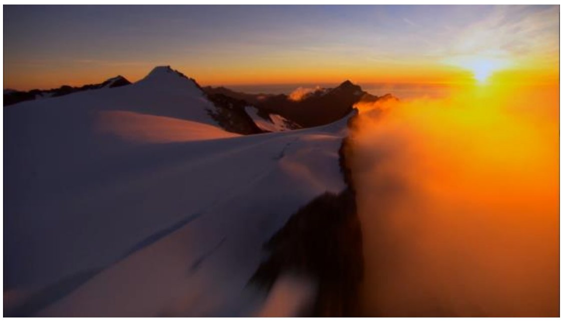
Before:

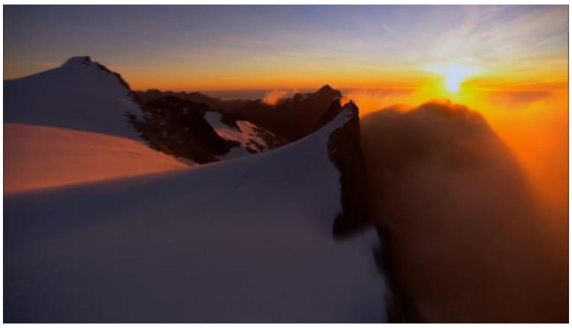


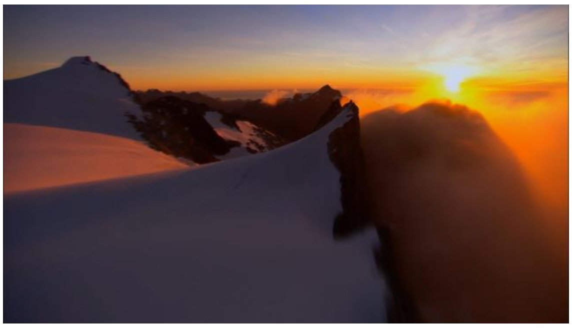
After:











6. Conclusion and Future Work

The proposed construction of video steganography was realized by embedding the secret image into the meaningful cover image of any type of video file using LSB approaches. A stego-key has been applied to the system to choose cover frames in which the embedment of the image is done and increase security. The proposed embedded video steganography has many specific advantages such as user friendliness, simple and effective process of embedding secret image with more security.

Future work we can do to improve this project are:

- Suitably encrypt data so as to match the cover file and hence make it more difficult to detect and decrypt hidden data.
- Modify our implementation techniques to make it faster and hence, satisfy suitable time constraints.
- Compressing the video file obtained by the stream of images with modified cover frames losslessly so as to increase the speed of transmission.
- Implement other higher order security measures so as to protect messages from steganalysis, cryptanalysis and various other types of sophisticated attacks.

7. References

1. S.Singh and G. Agarwal. *Hiding image into a video: A new approach of LSB replacement. (2010).*

2. Kamred Udham Singh. *Video Steganography: Text Hiding in Video by LSB substitution. (May 2014).*

3. James Ridgeway. *Video steganography (COM3600 Research Project*). (*April 2013).*