Image Steganography

Technical Report · December 2015

DOI: 10.13140/RG.2.2.21323.18727

Savitha Bhallamudi
Wright State University
2 PUBLICATIONS 0 CITATIONS

WRIGHT STATE UNIVERSITY

Department of Electrical Engineering

Image Steganography

Final project – Report

EE 7150 – Digital Image Processing

Fall 2015

Faculty: Dr. Arnab Shaw

Savitha Bhallamudi

Date due: Dec 18, 2015

Date handed in: Dec 18, 2015

ABSTRACT:

The project deals with learning about the various types of steganography available. Image steganography is performed for images and the concerning data is also decrypted to retrieve the message image. Since this can be done in several ways, image steganography is studied and one of the methods is used to demonstrate it.

Image steganography refers to hiding information i.e. text, images or audio files in another image or video files. The current project aims to use steganography for an image with another image using spatial domain technique. This hidden information can be retrieved only through proper decoding technique. This encryption and decryption of the images is done using MATLAB codes.

INTRODUCTION:

Data hiding is of importance in many applications. For hobbyists, secretive data transmission, for privacy of users etc. the basic methods are: Steganography and Cryptography.

Steganography is a simple security method. Generally there are three different methods used for hiding information: steganography, cryptography, watermarking.

In cryptography, the information to be hidden is encoded using certain techniques; this information is generally understood to be coded as the data appears nonsensical.

Steganography is hiding information; this generally cannot be identified because the coded information doesn’t appear to be abnormal i.e. its presence is undetectable by sight.

Detection of steganography is called Steganalysis.

Steganography is of different types:

1. Text steganography
2. Image steganography
3. Audio steganography
4. Video steganography

In all of these methods, the basic principle of steganography is that a secret message is to be embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So it cannot be detected easily to be containing hidden information unless proper decryption is used.

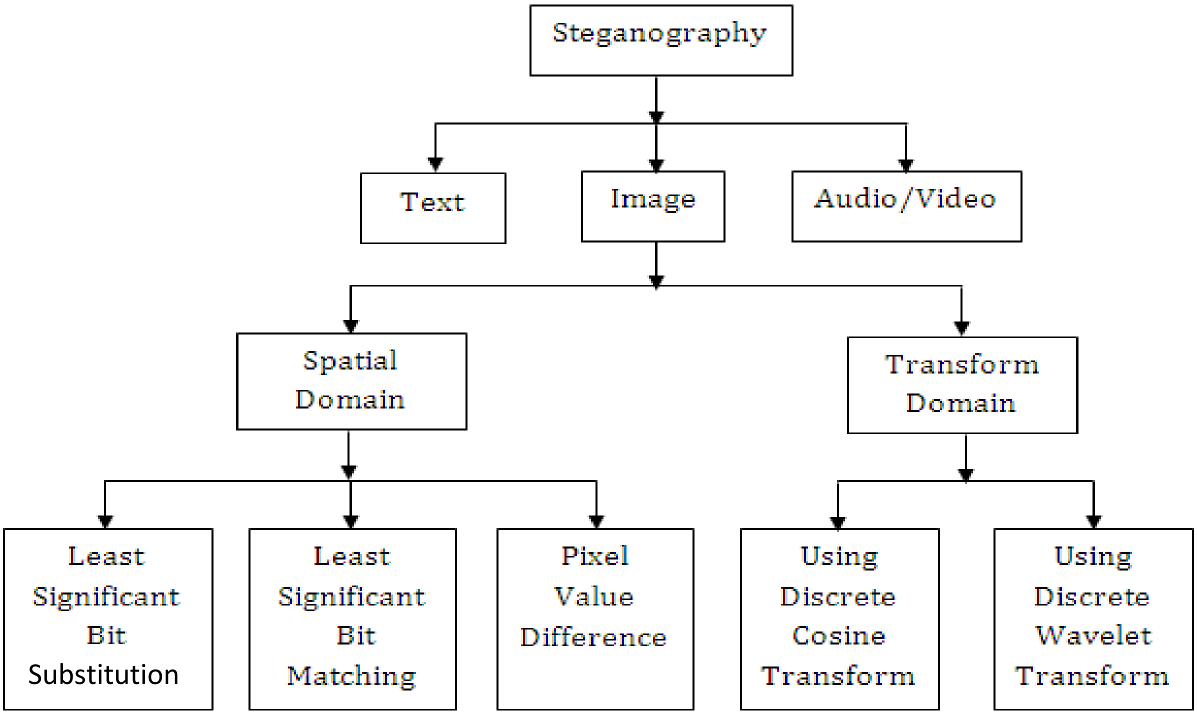

Steganography
Text
Image
Audio/Video
Spatial Domain
Transform Domain
Least Significant Bit Substitution
Least Significant Bit Matching
Pixel Value Difference
Using Discrete Cosine Transform
Using Discrete Wavelet Transform

Fig. 1. (Image source: 4.bp.blogspot.com/nFLojaiVBxA/Uq6_8Ji01PI/AAAAAAAAAzg/ruYCWjhBvZY/s1600/fig2.PNG)

As the above explanation goes, every steganography consists of three components:

1. Cover object
2. Message object
3. Resulting Steganographic object

In this project LSB substitution method is implemented and DCT method is discussed for image steganography. MATLAB is used for coding. The codes and result images are in the following report.

TECHNICAL DISCUSSION:

In the current project image steganography is dealt with using data hiding.

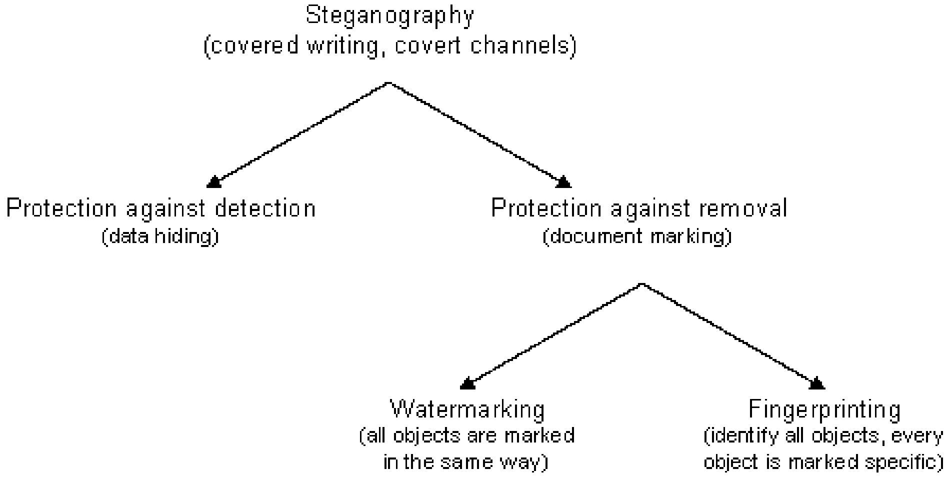

Steganography
(covered writing, covert channels)
Protection against detection
(data hiding)
Protection against removal
(document marking)
Watermarking
(all objects are marked in the same way)
Fingerprinting
(identify all objects, every object is marked specific)

Fig. 2. (Imagesource: www.cs.bham.ac.uk/~mdr/teaching/modules03/security/students/SS5/Steganography.htm)

There are two different methods for image steganography:

1. Spatial methods
2. Transform methods

In spatial method, the most common method used is LSB substitution method.

Least significant bit (LSB) method is a common, simple approach to embedding information in a cover file.

In steganography, LSB substitution method is used. I.e. since every image has three components (RGB). This pixel information is stored in encoded format in one byte. The first bits containing this information for every pixel can be modified to store the hidden text. For this, the preliminary condition is that the text to be stored has to be smaller or of equal size to the image used to hide the text.

LSB based method is a spatial domain method. But this is vulnerable to cropping and noise. In this method, the MSB (most significant bits) of the message image to be hidden are stored in the LSB (least significant bits) of the image used as the cover image.

It is known that the pixels in an image are stored in the form of bits. In a grayscale image, the intensity of each pixel is stored in 8 bits (1byte). Similarly for a colour (RGB-red, green, blue) image, each pixel requires 24 bits (8bits for each layer).

The Human visual system (HVS) cannot detect changes in the colour or intensity of a pixel when the LSB bit is modified. This is psycho-visual redundancy since this can be used as an advantage to store information in these bits and yet notice no major difference in the image.

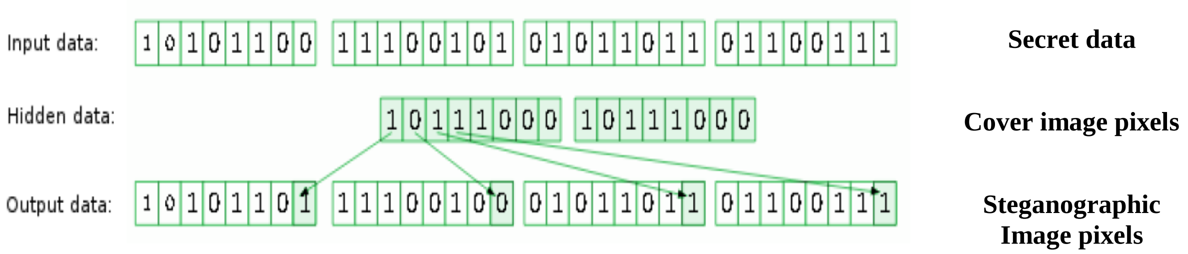


Fig. 3. (Image source: lvee.org/uploads/abstract_file/file/111/2.png)

<u>Steps used in LSB steganography:</u>

a. Steps for hiding message image:
   1. Read the image to be used as cover image. Noise is added to make it easier to disguise changes due to embedding the message image.
   2. Read the image to be sued as message image.
   3. Separate the bit planes of each image.

As it is known that the LSB (least significant bit) plane contains the least information associated with any image, and the MSB (most significant bit) plane contains most of the shape, colour information of an image.

It is generally ideal to replace up to 4 least bitplanes of the cover image, with the upper 4 bitplanes without revealing changes in the resultant image. Lesser number of bitplanes from the message image could be used, but the retrieved image would become distorted and loses information.

   4. Replace the least 4 bitplanes of cover image with the 4 most significant bitplanes from message image.
   5. Get the resultant Steganographic image by recombining these bitplanes.

b. Retrieving message image:
   1. Read the Steganographic image.
   2. Extract the required number of bitplanes of the image.
   3. Recombining the lower four bitplanes would give the retrieved message image.

Discrete Cosine Transform (DCT) method:

When information is embedded in spatial domain, losses can occur such as when the image is cropped etc. To overcome this problem the information is embedded in frequency domain in such a way that we embed the secret information in the significant frequency values and omit the higher frequency part. First the required transformations are applied and then accordingly to hide the secret message, the transform coefficients are changed.

Like in other transforms, decorrelation of the image data is required after applying discrete cosine transform (DCT). And encoding can be then done independently for each coefficient. Hence, compression efficiency is not lost.

In blocking method, blocks of the image are considered and DCT (discrete cosine transform) is done in order to break them. Each block is then subdivided into 64 parts (DCT coefficients). These coefficients are modified i.e. the colour gets modified a little by storing some text or another image in it. Embedding the secret data in the carrier image is generally done for the DCT coefficients that are lower than the chosen threshold value. But embedding information in DCT coefficient value 0 is avoided as this may lead to visual distortion of the cover image.

Palette modification:

In palette modification, the unused colours in an image's colour palette are replaced with colours to represent hidden message.

Palette Modification replaces the unused colours within an image's colour palette with colours that represent the hidden message.

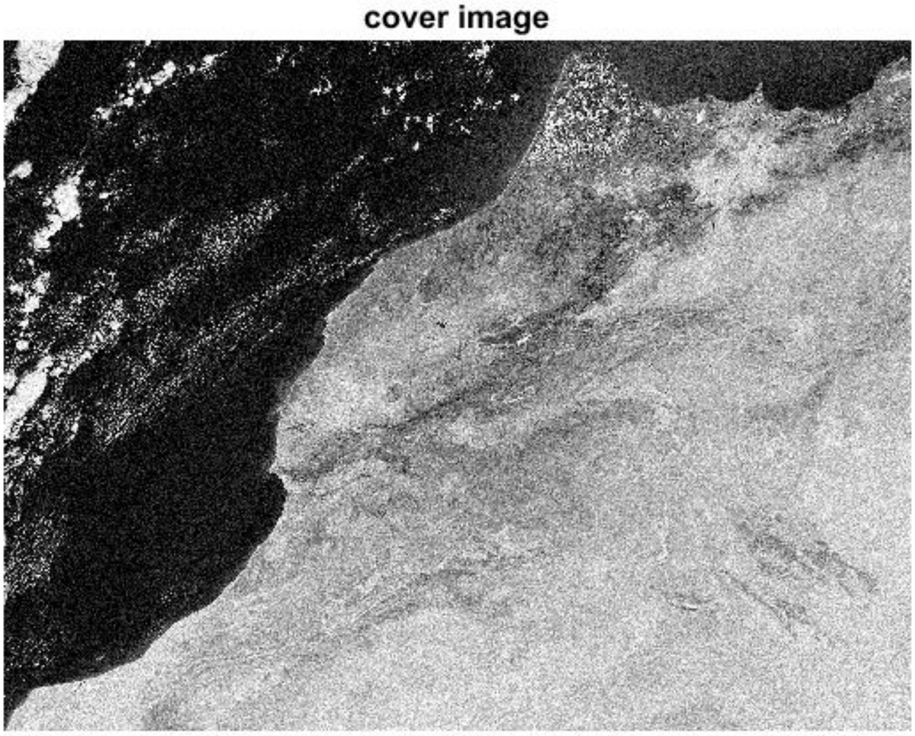
For example, we have an image containing 6shades of blue and 5 shades of brown. By modifying the bits, it is possible to generate a completely new palette of colours that were originally absent in the previous image. This changed colour palette may not be detected easily by human eye (HVS) and hence can be used to store other data or information.

- LSB technique can be used for BMP (bitmap) images. Since these involve lossless compression techniques.
- Blocking method – DCT and DWT – used for JPEG images. JPEG images have a lossy compression format, so spatial methods are unsuitable to perform steganography. DCT can be used to perform steganography on these images as, they undergo 2 layers of compression. One is lossless and then Huffman coding is used. The encryption data can be placed between these two layers.
- Palette based method – used for GIF images. GIF images have a very limited color palette. Therefore palette modification method is more suitable.

RESULTS:

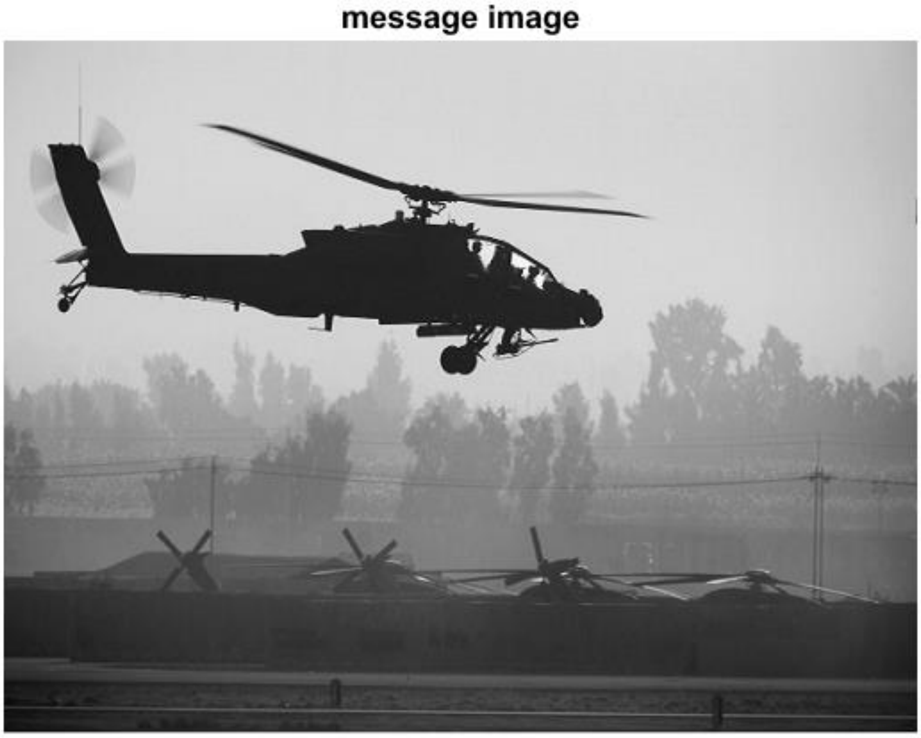
For Image Set 1:

Fig. 4: The original cover image used (converted to greyscale, with added noise):



(Image source: www.earthscienceworld.org/images/search/lightbox2.html?ID=h4vk4h)

Fig. 5: The secret message image to be hidden: (converted to greyscale)



(Image source: www.aircav.com/dodphoto/dod08/ah64-080103.html)

Fig. 6: The Steganographic image obtained after LSB steganography:

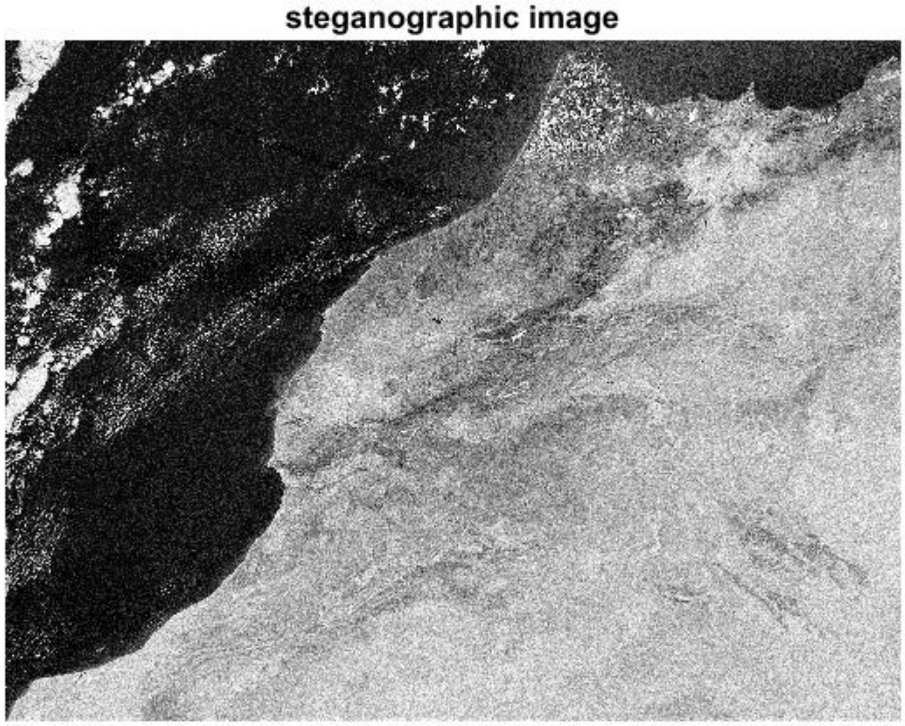


Fig. 7: The Recovered message image after LSB steganography:

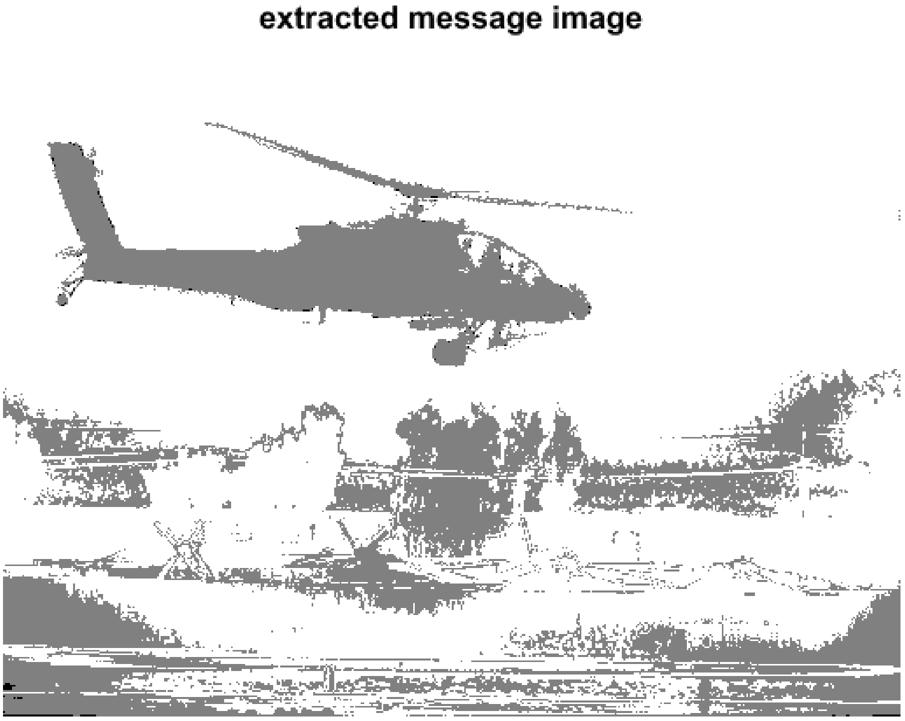
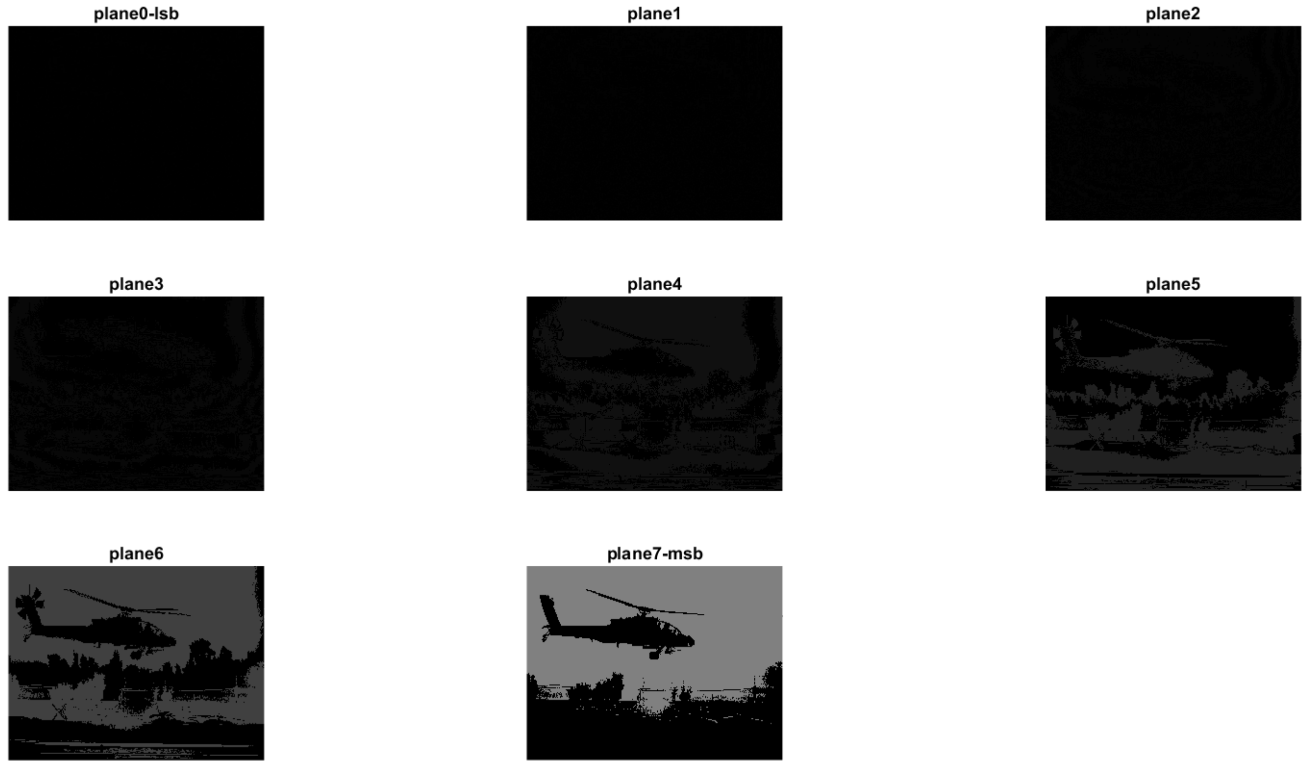
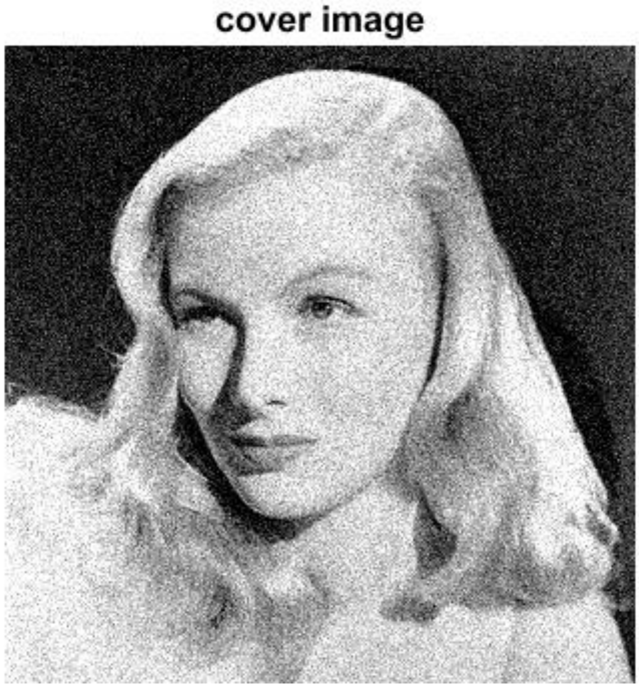


Fig. 8: The extracted bit-planes of message image:



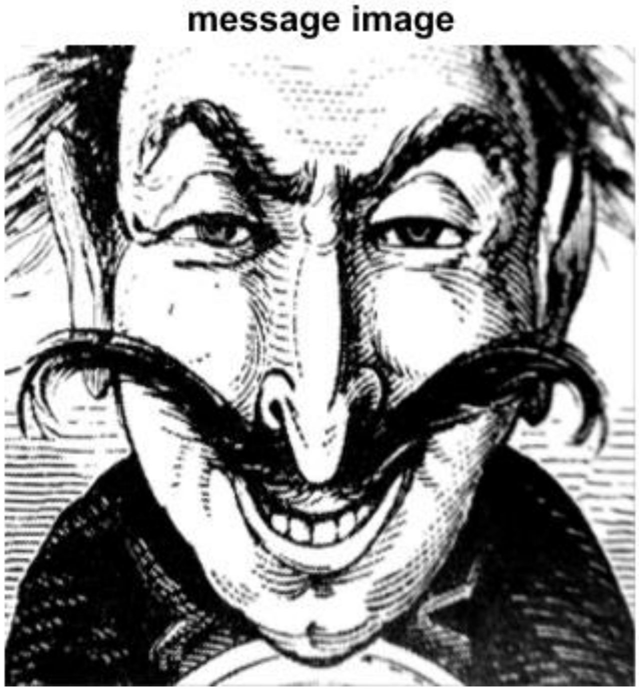
For Image Set 2:

Fig. 9: The original cover image used (converted to greyscale, with added noise):



(Image source: http://1.bp.blogspot.com/-IFlqcYhU6Wk/TgnbuQev2PI/AAAAAAAAACE/td-MWbAepis/s320/GRAYSCALE.colorize01.jpg (converted to .bmp for usage))

Fig. 10: The secret message image to be hidden: (converted to grey-scale)



(Image source: http://thelawlers.com/Blognosticator/wp-content/uploads/2017/02/Engraving-as-grayscale.jpg (resized and converted to .bmp for usage))

Fig. 11: The Steganographic image obtained after LSB steganography:

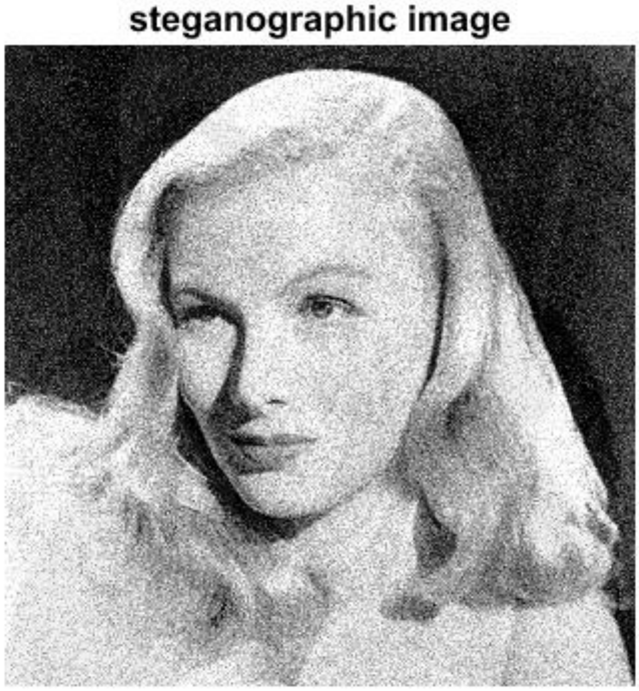


Fig. 12: The Recovered message image after LSB steganography:

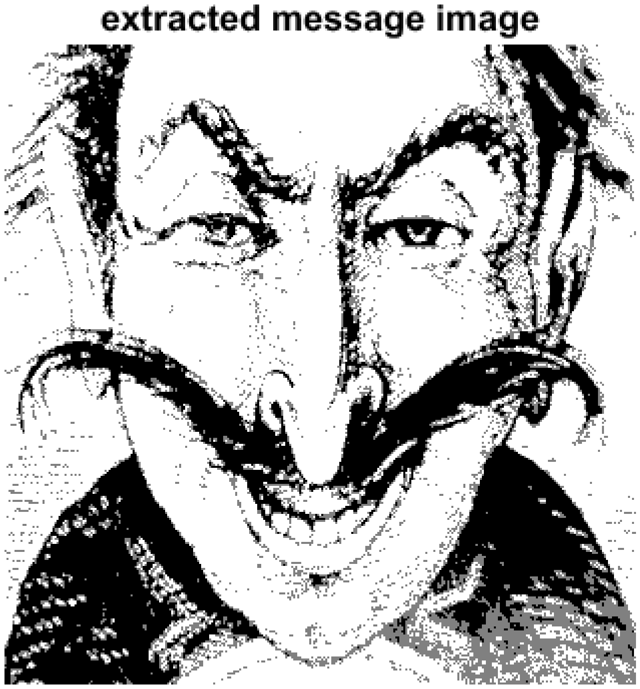
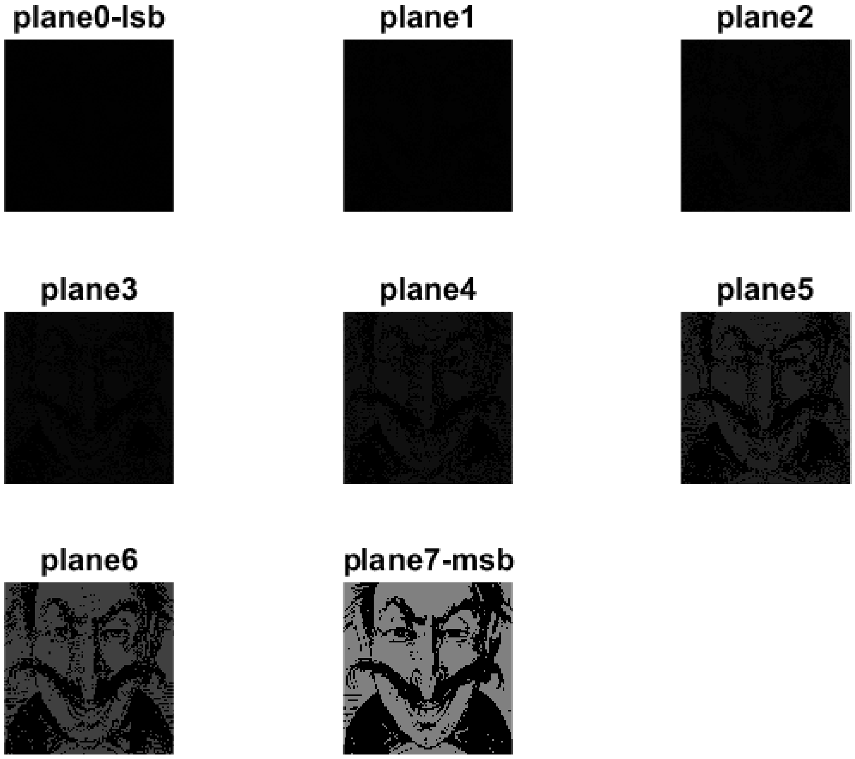


Fig. 13: The extracted bit-planes of message image:



CONCLUSION:

It is observed that through LSB Substitution Steganographic method, the results obtained in data hiding are pretty impressive as it utilizes the simple fact that any image could be broken up to individual bit-planes each consisting of different levels of information. It is to be noted that as discussed earlier, this method is only effective for bitmap images as these involve lossless compression techniques. Also, in this project grey-scale images have been used for demonstration. But this process can also be extended to be used for color images where, bit-plane slicing is to be done individually for the top four bit-planes for each of R, G, B of the message image, which are again to be placed in the R, G, B planes of the cover image, and extraction is done similarly.

It can be observed that the result image after extraction (Fig. 7) is not very clear as the initial message/secret image (Fig. 5). This can be explained by the fact that this is a very high resolution image. From Fig. 8, it is seen that data is visible in the message image in planes 5, 6, 7 to the human eye, and rest of the bit-planes appear to be dark/ black, but these also have tiny bits of information which is ignored for the process of data hiding. Better results can be obtained if the message image to be hidden is of a lower resolution. This is observed from Image set 2, where the extracted image (Fig. 12) has less loss compared to the original message image (Fig. 10).

It is also important to discuss that though steganography was once undetected, with the various methods currently used, it is not only easy to detect the presence but also retrieving them is easier. For instance, without having to use a software or complex tools for detection, simple methods to observe if an image file has been manipulated are:

1. Size of the image: A Steganographic image has a huge storage size when compared to a regular image of the same dimensions. I.e. if the original image storage size would be few KBs, the Steganographic image could be several MBs in size. This again varies with the resolution and type of image used.
2. Noise in image: A Steganographic image has noise when compared to a regular image. This is the reason why initially little noise is added to the cover image, so that the Steganographic image doesn’t appear very noisy when compared to the original cover image.

Though this project focusses on LSB and spatial domain steganography, few details about transform domain methods have also been researched, basics of which have been discussed. So through the various articles and theory available, it is observed that transform domain methods perform better in comparison with spatial domain methods.

APPENDIX:

The following codes were used to implement LSB steganography and also extract the message image:

```
1 ) %lsb steganography

clc;clear all;close all;

%cover image
cvr = imread('cover1.bmp');
% cvr = imread('cover2.bmp');
% cvr = im2double(cvr);
cvr = rgb2gray(cvr);
cvr=imnoise(cvr,'gaussian'); %adding gaussian noise
figure;imshow(cvr);title('cover image');

%mesage image
msg = imread('message1.bmp');
% msg = imread('message2.bmp');
% msg = im2double(msg);
msg = rgb2gray(msg);
figure;imshow(msg);title('message image');

m1=size(cvr,1);%height of cover image
n1=size(cvr,2);%width of cover image
m2=size(msg,1);%height of message image
n2=size(msg,2);%width of message image
% cvr = imcrop(cvr, [0 0 n2 m2]); %make dimensions equal
% figure;imshow(cvr);% [m3,n3]=size(cvr);
% disp(m1);disp(n1);disp(m2);disp(n2);disp(m3);disp(n3);

%extract upper 4 bitplanes of message
steg1=msg;
for i=1:m2
    for j=1:n2
        msg_b7(i,j)=bitand(steg1(i,j),128); %msb of message
        msg_b6(i,j)=bitand(steg1(i,j),64);
        msg_b5(i,j)=bitand(steg1(i,j),32);
        msg_b4(i,j)=bitand(steg1(i,j),16);
    end
end

%extract 8 bitplanes of cover image
steg2=cvr;
for i=1:m2
    for j=1:n2
        cvr_b7(i,j)=bitand(steg2(i,j),128); %msb of coverimage
        cvr_b6(i,j)=bitand(steg2(i,j),64);
        cvr_b5(i,j)=bitand(steg2(i,j),32);
        cvr_b4(i,j)=bitand(steg2(i,j),16);
        cvr_b3(i,j)=bitand(steg2(i,j),8);
        % bitand(steg2(i,j),4)=msg_b7;
        cvr_b2(i,j)=bitand(steg2(i,j),4);
        cvr_b1(i,j)=bitand(steg2(i,j),2);
        cvr_b0(i,j)=bitand(steg2(i,j),1); %lsb of coverimage
    end
end

%use last 4 layers to store message image
steg=zeros(size(cvr));%consider new blankimage
%msb of steg
steg=bitset(steg,8,cvr_b7); %msb of cover
steg=bitset(steg,7,cvr_b6);
steg=bitset(steg,6,cvr_b5);
steg=bitset(steg,5,cvr_b4); %bit-plane 4 of cover
steg=bitset(steg,4,msg_b7); %msb of message
steg=bitset(steg,3,msg_b6);
steg=bitset(steg,2,msg_b5);
%lsb of steg
steg=bitset(steg,1,msg_b4); %lsb of message
steg=uint8(steg);
figure,imshow(steg);title('steganographic image');
imwrite(steg,'steganographic1.bmp');
% imwrite(steg,'steganographic2.bmp');
```

---

2) %extracting message - lsb steganography

```
clc;clear all;close all;

img=imread('steganographic1.bmp');
% img=imread('steganographic2.bmp');
figure;imshow(img);title('steganographic image');
[m,n]=size(img);

for i=1:m
    for j=1:n
        img_b7(i,j)=bitand(img(i,j),128); %msb of image
        img_b6(i,j)=bitand(img(i,j),64);
        img_b5(i,j)=bitand(img(i,j),32);
        img_b4(i,j)=bitand(img(i,j),16);
        img_b3(i,j)=bitand(img(i,j),8);
        % bitand(steg2(i,j),4)=msg_b7;
        img_b2(i,j)=bitand(img(i,j),4);
        img_b1(i,j)=bitand(img(i,j),2);
        img_b0(i,j)=bitand(img(i,j),1); %lsb of image
    end
end

message = img_b0*128 + img_b1*64 + img_b2*32 + img_b3*16;
figure;imshow(message);title('extracted message image');
imwrite(message,'retrieved_lsb1.bmp');
%imwrite(message,'retrieved_lsb2.bmp');

% figure;
% subplot(4,4,1);imshow(img3_b0);title('lsb');
% subplot(4,4,2);imshow(img3_b1);
% subplot(4,4,3);imshow(img3_b2);
% subplot(4,4,4);imshow(img3_b3);
% subplot(4,4,5);imshow(img3_b4);
% subplot(4,4,6);imshow(img3_b5);
% subplot(4,4,7);imshow(img3_b6);
% subplot(4,4,8);imshow(img3_b7);title('msb');

% ext=zeros(size(img));
% ext=bitset(ext,8,bitget(img,4));
% ext=bitset(ext,7,bitget(img,3));
% ext=bitset(ext,6,bitget(img,2));
% ext=bitset(ext,5,bitget(img,1));
% figure,imshow(ext);
% imwrite(steg,'4layers.bmp');
```

3)

```
%bitplane extraction and display - message image
clc;clear all;close all;

%mesage image
msg = imread('message1.bmp');
% msg = imread('message2.bmp');
% msg = im2double(msg);
msg = rgb2gray(msg);

h=size(msg,1);%height of message image
w=size(msg,2);%width of message image

%extract 8 bitplanes of message
img1=msg;
for i=1:h
    for j=1:w
        msg_b7(i,j)=bitand(img1(i,j),128); %msb of message
        msg_b6(i,j)=bitand(img1(i,j),64);
        msg_b5(i,j)=bitand(img1(i,j),32);
        msg_b4(i,j)=bitand(img1(i,j),16);
        msg_b3(i,j)=bitand(img1(i,j),8);
        msg_b2(i,j)=bitand(img1(i,j),4);
        msg_b1(i,j)=bitand(img1(i,j),2);
        msg_b0(i,j)=bitand(img1(i,j),1); %lsb of message
    end
end

figure;
subplot(3,3,1);imshow(msg_b0);title('plane0-lsb');
subplot(3,3,2);imshow(msg_b1);title('plane1');
subplot(3,3,3);imshow(msg_b2);title('plane2');
subplot(3,3,4);imshow(msg_b3);title('plane3');
subplot(3,3,5);imshow(msg_b4);title('plane4');
subplot(3,3,6);imshow(msg_b5);title('plane5');
subplot(3,3,7);imshow(msg_b6);title('plane6');
subplot(3,3,8);imshow(msg_b7);title('plane7-msb');
```

REFERENCES:

1. Johnson, Neil F. "Steganography." *Http://www.jjtc.com/pub/tr_95_11_nfj/sec101.html*. N.p., Nov. 1995. Web.
2. Shikha, and Vidhu Kiran Dutt. "International Journal of Advanced Research in Computer Science and Software Engineering." *Http://www.ijarcsse.com/*. N.p., Sept. 2014. Web.
3. Niels Provos, and Peter Honeyman. “Hide and Seek: An Introduction to Steganography.”
   IEEE Security & Privacy Magazine, May-June 2013. Web.
4. Nick Nabavian. “Image steganography” Nov. 28, 2007. http://www1.chapman.edu/~nabav100/ImgStegano/download/ImageSteganography.pdf
5. Dr Ekta Walia, Payal Jain and Navdeep. “An analysis of LSB & DCT based Steganography.” Global Journal of Computer Science and Technology, April2010. https://globaljournals.org/GJCST_Volume10/gjcst_vol10_issue_1_paper8.pdf
6. D'Incau, Paolo. "LSB watermarking using MATLAB | Paolo D'Incau's Blog." *Paolo D'Incau's Blog | Introduction to erlang and other useful stuff.* 22 Mar 2010. Web. 5 Dec 2015. pdincau.wordpress.com/2010/03/22/lsb-watermarking-using-matlab/.
7. Hardik Patel, and Preeti Dave. "International Journal of Advanced Research in Computer Science and Software Engineering." *Http://www.ijarcsse.com/*. N.p., Jan-Feb. 2012. Web.
8. Deepak Singla, and Rupali Syal. "International Journal of Computational Engineering Research." *Citeseerx.ist.psu.edu*. N.p., Mar-Apr. 2012. Web.
9. Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn, Information Hiding-A Survey, IEEE, special issue on protection of multimedia content, Jul 1999, 1062-1078.
10. Analyst, Image. “Re: How to Change least Significant Binary Bits of Each Bit of Cover Image by Other Four Binary Bits??” Blog comment. *MATLAB Central*. N.p., 29 Jan. 2013. Web. 5 Dec. 2015.
    www.mathworks.com/matlabcentral/answers/60261-how-to-change-least-significant-4-binary-bits-of-each-bit-of-cover-image-by-other-four-binary-bits#answer_72801
11. Balajee, J. “MATLAB CODING’s – PART II.” researchviews.blogspot.com. N.p., 21 Nov. 2012. Web. 12 Dec. 2015.
    http://researchviews.blogspot.com/2012/11/matlab-codings-part-ii.html
12. G.A. Papakostas, D.E. Koulouriotis and E.G. Karakasis (2009). Efficient 2-D DCT Computation from an Image Representation Point of View, Image Processing, Yung-Sheng Chen (Ed.), ISBN: 978-953-307-026-1, InTech,
    www.intechopen.com/books/image-processing/efficient-2-d-dct-computation-from-animage-representation-point-of-view
13. Andrew B. Watson. “Image Compression Using the Discrete Cosine Transform.” Mathematica Journal, 4(1), 1994, p. 81-88
    www.vision.arc.nasa.gov/publications/mathjournal94.pdf

14. Help for functions in Command Window - MATLAB help. www.mathworks.com/help/matlab/ref/help.html. Accessed 5 Dec. 2015.
15. Monika Kwiatkowska, and Lukasz Swierczewski, "Steganography - coding and intercepting the information from encoded pictures in the absence of any initial information." Linux Vacation / Eastern Europe (LVEE 2014), N.p., 21 Feb. 2014. Web. 10 Dec. 2015.
lvee.org/en/abstracts/106
16. Nichal, Arjun. "MATLAB Implementation of Steganography (Simple Data Hiding Method)." *Image Processing Fundamentals, Basics of MATLAB and Embedded System Practicals on LPC2148....* N.p., 01 Jan. 1970. Web. 7 Dec. 2015. https://imagelpcmatlab.blogspot.com/2013/12/matlab-implementation-of-steganography.html
17. Eloise. "AP 186 Blog Reports: Image Types and Formats." *AP 186 Blog Reports*, 28 June 2011, jigglingatoms.blogspot.com/2011/06/image-types-and-formats.html. Accessed 8 Dec. 2015.