

|  |
| --- |
| **Title:** Implementation of Steganography using LSB algorithm. |

**Objective:** To study steganography using bit plane slicing

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **CO** | **Outcome** |
| **CO4** | Design & implement algorithms for digital image enhancement, segmentation & restoration. |

**Books/ Journals/ Websites referred:**

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html.
3. R. C.Gonsales R.E.Woods, “Digital Image Processing”, Second edition, Pearson Education
4. S.Jayaraman, S Esakkirajan, T Veerakumar “Digital Image Processing “Mc Graw Hill.
5. S.Sridhar,”Digital Image processing”, oxford university press, 1st edition."

**Pre Lab/ Prior Concepts:**

Image enhancement techniques in spatial domain

1. Bit plane slicing.

Steganography is the art and science of communicating in a way which hides the existence of the communication. Steganography plays an important role in information security. It is the art of invisible communication by concealing information inside other information. The term steganography is derived from Greek and literally means “covered writing”. A Steganography system consists of three elements: cover image (which hides the secret message), the secret message and the stego-image (which is the cover object with message embedded inside it).

A digital image is described using a 2-D matrix of the colour intestines at each grid point (i.e. pixel). Typically gray images use 8 bits, whereas coloured utilizes 24 bits to describe the colour model, such as RGB model. The Steganography system which uses an image as the cover, there are several techniques to conceal information inside cover-image. The spatial domain techniques manipulate the cover-image pixel intensity values to embed the secret information. The secret bits are written directly to the cover image pixel intensity bytes by changing least significant bit. Consequently, the spatial domain techniques are simple and easy to implement. The Least Significant Bit (LSB) is one of the conventional techniques in spatial domain image Steganography. The LSB based image steganography embeds the secret information in the least significant bits of pixel values of the cover image.

**Encoder:** To hide text message into an image (cover)

Step 1: Accept small secrete text message and cover image.

Step 2: Convert secrete message into series of bits.

Step 3: Read a pixel from image and read message bit, to hide that bit into Least Significant Bit of pixel intensity.

Step 4: Continue step 3 until all secrete message bits are not hidden in the image.

Step 5: save new image as stego-image.

**Decoder:** To retrieve message from stego-image.

Step 1: Accept stego-image.

Step 2: Read the pixel intensity from stego-image and convert the intensity value in binary representation

Step 3: Read LSB bit of this binary value and save it in text file.

Step 4: Continue step 2 and 3 until all message bits are extracted.

Step 5: Now convert message bits into original text message and display this message.

**Implementation Details:**

first=imread('img.jpg');

first=rgb2gray(first);

subplot(3,3,1)

imshow(first)

title('First Original image');

second=imread('ball.jpg');

second=rgb2gray(second);

subplot(3,3,3)

imshow(second)

title('Second Original image');

second = bitand(second,128);

first = bitand(first,254) + second/128;

subplot(3,3,4)

imshow(second)

title('MSB of second image');

subplot(3,3,6)

imshow(first)

title('Encoded image');

final = bitand(second,0);

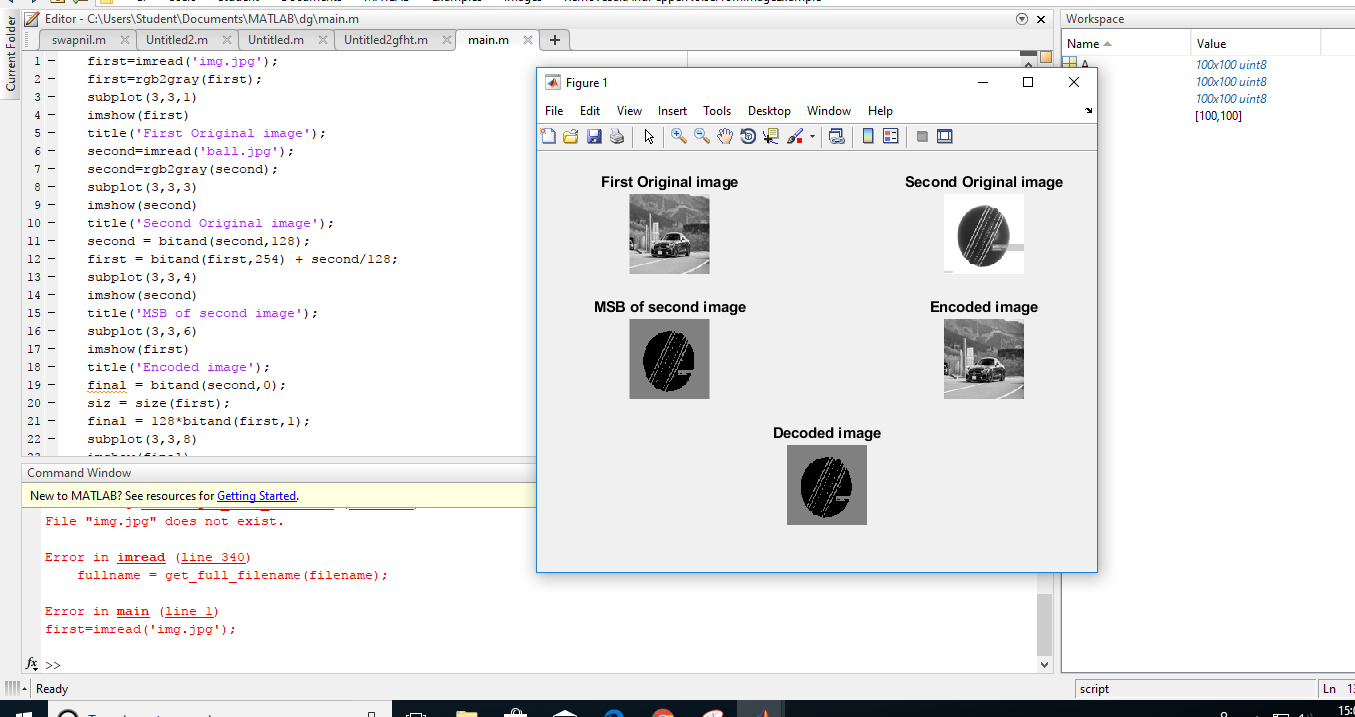
siz = size(first);

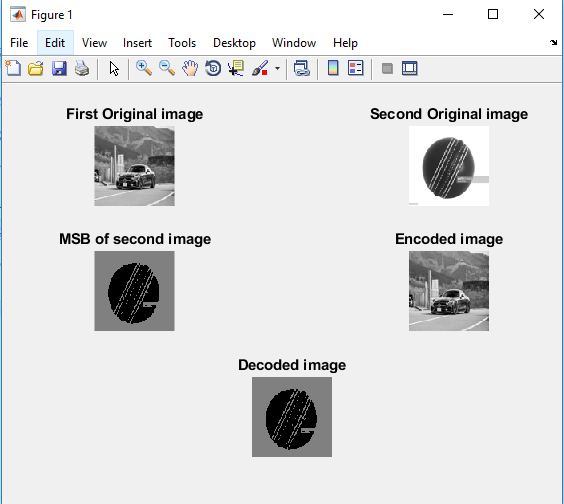
final = 128\*bitand(first,1);

subplot(3,3,8)

imshow(final)

title('Decoded image');

****

****

**Write Algorithm and Matlab commands used:**

**FOR MSB**

first = bitand(first,254) + second/128;

**FOR ENCODED**

final = 128\*bitand(first,1);

**Conclusion:**

Steganography is one of the most powerful techniques to conceal the information inside a cover. The cover can be of any type such as image, audio, video, text and information can be such as image, audio, video, text. In this experiment text data is hidden inside an image. So it’s a covert way of communication.

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

**Post Lab Descriptive Questions**

1. Explain the need of LOG operator.

A logarithm is a mathematical operation that determines how many times a certain number, called the base, is multiplied by itself to reach another number. Because logarithms relate geometric progressions to arithmetic progressions, examples are found throughout nature and art, such as the spacing of guitar frets, mineral hardness, and the intensities of sounds, stars, windstorms, earthquakes and acids. Logarithms even describe how humans instinctively think about numbers

The dynamic range of an image can be compressed by replacing each [pixel value](https://homepages.inf.ed.ac.uk/rbf/HIPR2/value.htm) with its logarithm. This has the effect that low intensity pixel values are enhanced. Applying a pixel logarithm operator to an image can be useful in applications where the dynamic range may too large to be displayed on a screen (or to be recorded on a film in the first place).

The most common application for the dynamic range compression is for the display of the [Fourier Transform](https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm). We will illustrate this using



The maximum magnitude value of its Fourier Transform is Eqn:eqnpxlg4, and the second largest value is approximately 10 times smaller. If we simply linearly [scale](https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixmult.htm) this image, we obtain

cln1fur1

Due to the large dynamic range, we can only recognize the largest value in the center of the image. All remain values appear as black on the screen. If we instead apply the logarithmic operator to the Fourier image, we obtain



Here, smaller pixel values are enhanced and therefore the image shows significantly more details.

The logarithmic operator enhances the low intensity pixel values, while compressing high intensity values into a relatively small pixel range. Hence, if an image contains some important high intensity information, applying the logarithmic operator might lead to loss of information. For example,

stp1fur1

is the linearly [scaled](https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixmult.htm) Fourier Transform of



The image shows one bright spot in the center and two darker spots on the diagonal. We can infer from the image that these three frequencies are the main components of the image with the DC-value having the largest magnitude. Applying the logarithmic transform to the Fourier image yields



Here, we can see that the image contains many more frequencies. However, it is now hard to tell which are the dominating ones, since all high magnitudes are compressed into a rather small pixel value range. The magnitude of compression is large in this case because there are extremely high intensity values in the output of the Fourier Transform (in this case up to Eqn:eqnpxlg5). We can decrease the compression rate by [scaling](https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixmult.htm) down the Fourier image before applying the logarithmic transform. Image

stp1fur6

is the result of first multiplying each pixel with 0.0001 and then taking its logarithm. Now, we can recognize all the main components of the Fourier image and can even see the difference in their intensities.

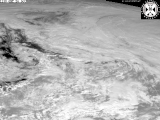
Thus, a logarithmic transform is appropriate when we want to enhance the low pixel values at the expense of loss of information in the high pixel values. For example, the man in



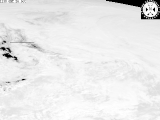
was photographed in front of a bright background. The dynamic range of the film material is too small, so that the graylevels on the subject's face are clustered in a small pixel value range. A logarithmic transform spreads them over a wider range, while the higher values are compressed. The result can be seen in



On the other hand, applying a logarithmic transform to



is less appropriate, because most of its details are contained in the high pixel values. Applying the logarithmic operator yields

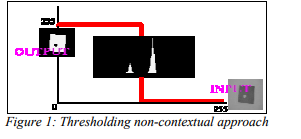


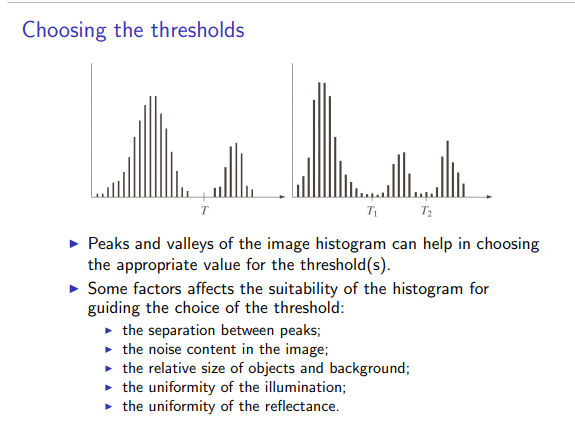
This image shows that a lot of information is lost during the transform.

1. Explain the technique of thresholding for segmentation.

Thresholding is the simplest method of image segmentation. From a gray scale image, thresholding can be used to create binary images. Binary images are produced from color images by segmentation. Segmentation is the process of assigning each pixel in the source image to two or more classes. If there are more than two classes then the usual result is several binary images. In image processing, thresholding is used to split an image into smaller segments, or junks, using at least one color or gray scale value to define their boundary. The advantage of obtaining first a binary image is that it reduces the complexityof the data and simplifies the process of recognition and classification. The most common way to convert a gray level image to a binary image is to select a single threshold value

Segmentation by thresholding I Thresholding is the simplest segmentation method. I The pixels are partitioned depending on their intensity value. I Global thresholding, using an appropriate threshold T: g(x, y) = 1, if f (x, y) > T 0, if f (x, y) ≤ T I Variable thresholding, if T can change over the image. I Local or regional thresholding, if T depends on a neighborhood of (x, y). I adaptive thresholding, if T is a function of (x, y). I Multiple thresholding: g(x, y) = a, if f (x, y) > T2 b, if T1 < f (x, y) ≤ T2 c, if f (x, y) ≤ T1





There are three types of thresholding algorithms.

* Global thresholding
* Local thresholding
* Adaptive thresholding

In adaptive thresholding, different threshold values for different local areas are used.