

|  |
| --- |
| **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:**

**Code**

a = imread('drag.jpg');

figure(1);

imshow(a);

cover = rgb2gray(a);

figure(2);

imshow(cover);

b = imread('mg.jpg');

figure(3);

imshow(b);

data = rgb2gray(b);

figure(4);

imshow(data);

[m,n]= size(data);

cover = imresize(cover,[m n]);

figure(2);

imshow(cover);

add = data;

bit = data;

toadd = data;

for k = 1:8 %extracting the msb of data image

for i = 1:m

for j = 1:n

if(k==8)

toadd(i,j) = mod(data(i,j),2);

data(i,j) = (data(i,j)-toadd(i,j))/2;

else

temp = mod(data(i,j),2);

data(i,j) = (data(i,j)-temp)/2;

end

end

end

end

mix = data;

for k = 1:8 %setting lsb of the cover image to the extracted msb of data image

figure(5);

for i = 1:m

for j = 1:n

if(k==1)

bit(i,j) = toadd(i,j);

cover(i,j) = (cover(i,j)-bit(i,j))/2;

else

bit(i,j) = bit(i,j)+(mod(cover(i,j),2)\*pow2(k-1));

cover(i,j) = (cover(i,j)-mod(cover(i,j),2))/2;

end

end

end

subplot(2,4,k),imshow(bit);

end

figure(6);

imshow(bit);

added = bit;

h=added;

for i = 1:m // extracting data image from cover image

for j = 1:n

h(i,j) = mod(added(i,j),2);

if(h(i,j)==1)

h(i,j)=128;

end

end

end

data = h;

figure(7);

imshow(data)

**Output**

1. **Cover Image**



1. **Grayscale Cover Image (after resizing according to data image)**



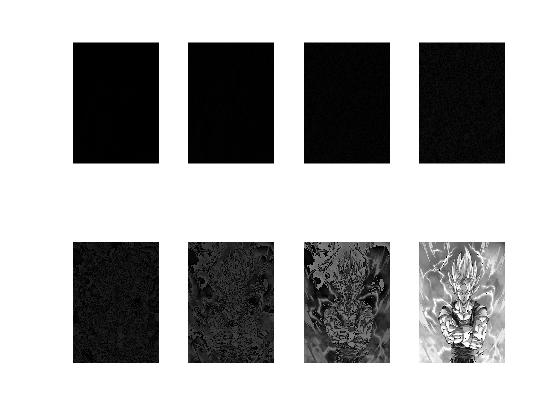
1. **Data Image**



1. **Grayscale Data Image**



1. **Bit planes after replacing the LSB of cover image with the MSB of data image (from LSB to MSB)**



1. **Image after replacing the LSB of cover image with the MSB of data image**



1. **Extracted data image from the cover image**



**Conclusion**

Thus, we have implemented Steganography using LSB algorithm.

**Date: 20/03/2019 Signature of faculty in-charge**

**Post Lab Descriptive Questions**

1. **Explain the need of LOG operator.**

Ans. 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.

The dynamic range of an image can be compressed by replacing each pixel value 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).

The maximum magnitude value of its Fourier Transform is 7.9 x 106, and the second largest value is approximately 10 times smaller. If we simply linearly scale 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.

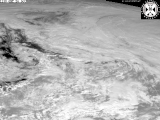
Thus, a logarithmic transform is appropriate when we want to enhance the low pixel values at the expense of loss of information in the high pixel values. For example, the man in:



was photographed in front of a bright background. The dynamic range of the film material is too small, so that the gray levels 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

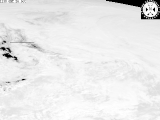


On the other hand, applying a logarithmic transform to:



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

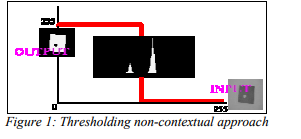
Applying the logarithmic operator yields:



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

1. **Explain the technique of thresholding for segmentation.**

Ans. Thresholding is the simplest method of image segmentation. From a grayscale 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 complexity of 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 T.



The input to a thresholding operation is typically a grayscale or color image. In the simplest implementation, the output is a binary image representing the segmentation. Black pixels correspond to background and white pixels correspond to foreground (or vice versa). This method of segmentation applies a single fixed criterion to all pixels in the image simultaneously.

**Image Segmentation = divide image into (continuous) regions or sets of pixels. The pixels are partitioned depending on their intensity value.**

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.