

| **Title:** Implement the following point processing techniques in spatial domain:   * Image Negative. * Thresholding. * Gray level slicing with and without background * Bit plane slicing |
| --- |

**Objective:** To learn & understand point processing techniques.

**Expected Outcome of Experiment:**

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

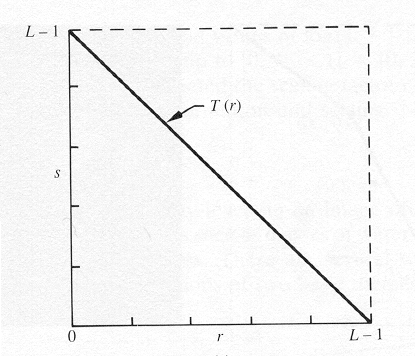
**Books/ Journals/ Websites referred:**

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 Negative:**

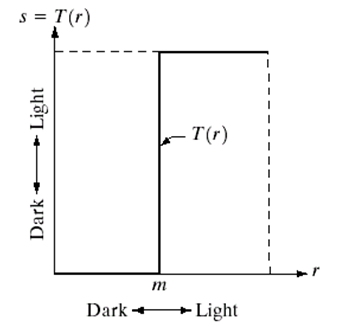
Negative images are useful for enhancing white or grey detail embedded in dark regions of an image. Image negatives are obtained by using the transformation function s=T(r).



[0,L-1] is the range of gray levels

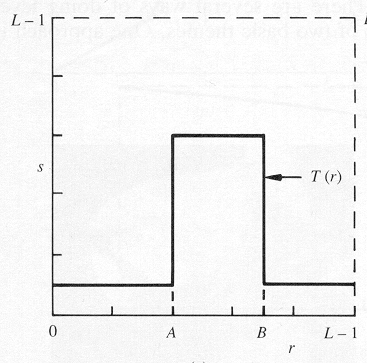
*S= L-*1*-r*

**Thresholding**

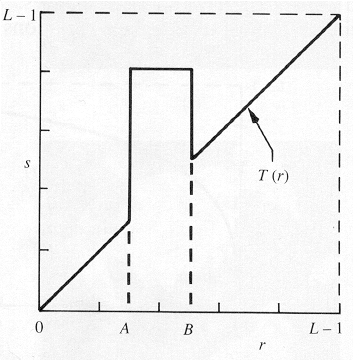
From a [grayscale](https://en.wikipedia.org/wiki/Grayscale) image, thresholding can be used to create [binary images](https://en.wikipedia.org/wiki/Binary_image). The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity  is less than some fixed constant T or a white pixel if the image intensity is greater than that constant. ****

**Gray Level Slicing**

To highlight a specific range of gray levels in an image (e.g. to enhance certain features). One way is to display a high value for all gray levels in the range of interest and a low value for all other gray levels (binary image).

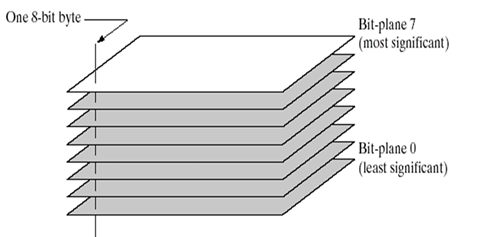


The second approach is to brighten the desired range of gray levels but preserve the background and gray-level tonalities in the image:



**Bit plane slicing**

Bit plane slicing is used to highlight the contribution made to the total image appearance by specific bits. Assuming that each pixel is represented by 8 bits, the image is composed of 8 1-bit planes. Plane 0 contains the least significant bit and plane 7 contains the most significant bit. Only the higher order bits (top four) contain visually significant data. The other bit planes contribute the more subtle details. Plane 7 corresponds exactly with an image thresholded at gray level 128.



|  |
| --- |

**Implementation steps with screenshots:**

**1] Image Negative:**

% Image Negative

I = imread('cameraman.png');

% Check if the image is already grayscale or RGB

if size(I, 3) == 3

I\_gray = rgb2gray(I);

else

I\_gray = I;

end

% Image Negative

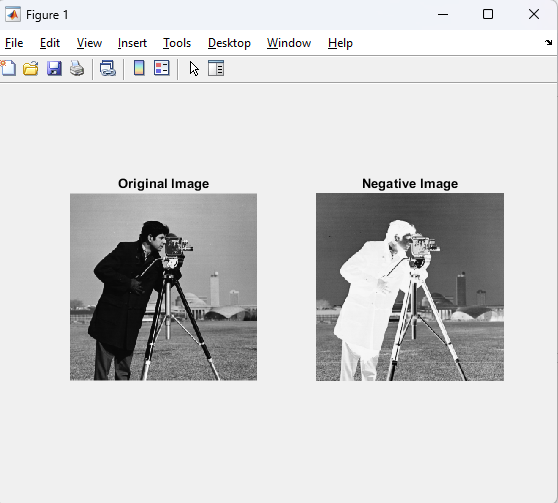
I\_negative = 255 - I\_gray;

figure;

subplot(1,2,1), imshow(I\_gray), title('Original Image');

subplot(1,2,2), imshow(I\_negative), title('Negative Image');

**OUTPUT:**

****

**2] Thresholding:**

I = imread('cameraman.png');

if size(I, 3) == 3

I\_gray = rgb2gray(I);

else

I\_gray = I;

end

threshold = 128;

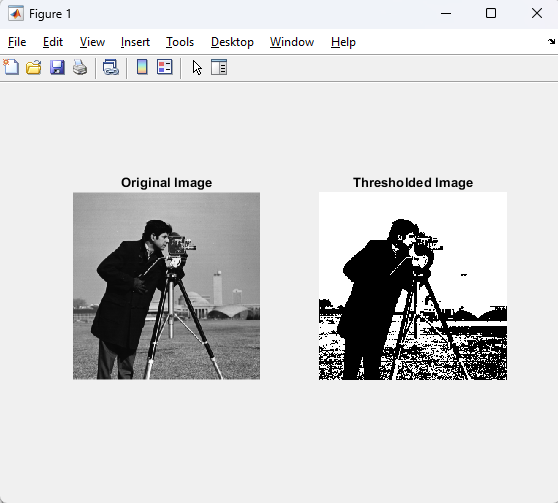
I\_thresholded = I\_gray > threshold;

figure;

subplot(1,2,1), imshow(I\_gray), title('Original Image');

subplot(1,2,2), imshow(I\_thresholded), title('Thresholded Image');

**OUTPUT:**

****

**3A] Gray Level Slicing with Background:**

**I = imread('cameraman.png');**

**if size(I, 3) == 3**

**I\_gray = rgb2gray(I);**

**else**

**I\_gray = I;**

**end**

**L1 = 100;**

**L2 = 150;**

**I\_sliced = I\_gray;**

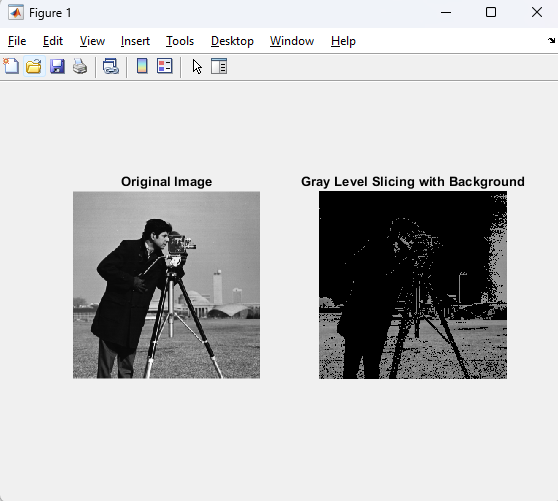
**I\_sliced(I\_gray < L1 | I\_gray > L2) = 0;**

**figure;**

**subplot(1,2,1), imshow(I\_gray), title('Original Image');**

**subplot(1,2,2), imshow(I\_sliced), title('Gray Level Slicing with Background');**

**OUTPUT:**

****

**3B] Gray Level Slicing without Background:**

**I = imread('cameraman.png');**

**if size(I, 3) == 3**

**I\_gray = rgb2gray(I);**

**else**

**I\_gray = I;**

**end**

**L1 = 100;**

**L2 = 150;**

**I\_sliced = I\_gray;**

**I\_sliced(I\_gray >= L1 & I\_gray <= L2) = 255;**

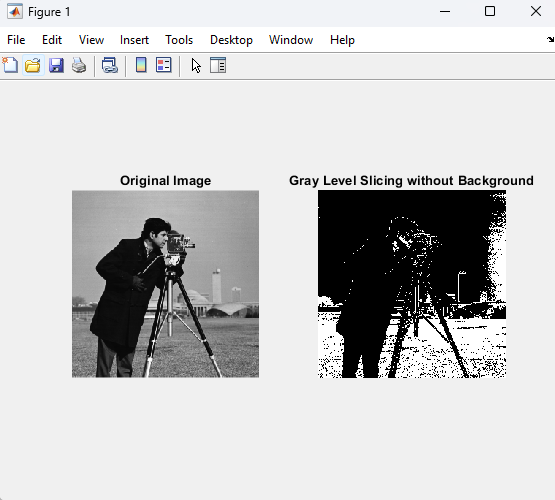
**I\_sliced(I\_gray < L1 | I\_gray > L2) = 0;**

**figure;**

**subplot(1,2,1), imshow(I\_gray), title('Original Image');**

**subplot(1,2,2), imshow(I\_sliced), title('Gray Level Slicing without Background');**

**OUTPUT:**

****

**4] Bit Plane Slicing:**

**I = imread('cameraman.png');**

**if size(I, 3) == 3**

**I\_gray = rgb2gray(I);**

**else**

**I\_gray = I;**

**end**

**bit\_planes = zeros(size(I\_gray, 1), size(I\_gray, 2), 8);**

**for i = 1:8**

**bit\_planes(:,:,i) = bitget(I\_gray, i);**

**end**

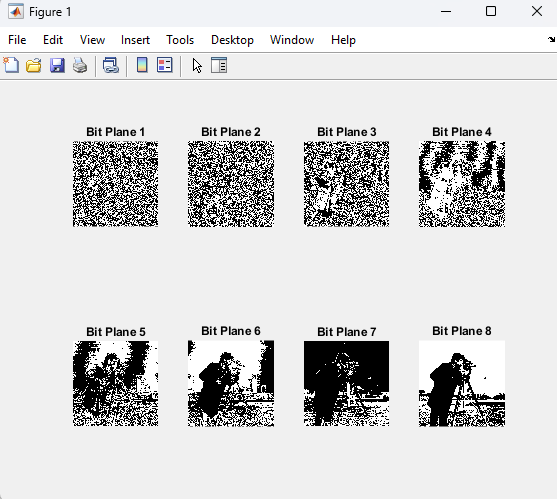
**figure;**

**for i = 1:8**

**subplot(2,4,i), imshow(bit\_planes(:,:,i)), title(['Bit Plane ', num2str(i)]);**

**end**

**OUTPUT:**

****

**Conclusion:-**

The experiment demonstrates key point processing techniques for image enhancement, enabling better understanding of image manipulation and segmentation.

**Date: 07-02-2025 Signature of faculty in-charge**

**Post Lab Descriptive Questions**

1. Explain the role of bit plane slicing in achieving Steganography concept.

**Bit Plane Slicing** is a method used to break down an image into its individual bit planes. In the context of **steganography**, this technique plays a critical role by allowing you to manipulate the least significant bits (LSBs) of an image without significantly affecting its visual appearance.

#### Steganography Overview:

Steganography is the practice of hiding information (like a secret message) within an image or other digital media. The goal is to embed information in such a way that it is imperceptible to the human eye.

#### Bit Plane Slicing in Steganography:

* **Bit planes** refer to the individual bits that make up a pixel’s value. For an 8-bit grayscale image, each pixel has 8 bits (ranging from 0 to 255). In a 16-bit image, each pixel has 16 bits, and so on.
* **Least Significant Bit (LSB)**: The least significant bit of a pixel is the last bit (rightmost bit) in its binary representation. The LSB holds the smallest amount of information and, when altered, minimally affects the overall intensity of the pixel.
* **Using LSB for Steganography**: The most common steganographic technique involves modifying the LSBs of an image's pixels to hide secret data (like a message or another image). Since the LSBs contribute minimally to the overall pixel value, changing them does not cause noticeable distortion to the image.

#### How Bit Plane Slicing is Used:

* **Embedding Information**: You can manipulate the LSBs (or even the second least significant bit) of an image to store secret information. By altering the LSBs of many pixels, you can embed a binary message without visually changing the image.
* **Extraction**: To extract the hidden message, you would use bit plane slicing to isolate the specific bit plane where the secret data is embedded and then reconstruct the message by extracting the bits.

#### Example:

* **Original Pixel (8-bit)**: 10011011 (binary)
* **LSB Modification**: By changing the least significant bit, you can embed a new bit of information without changing the pixel value drastically.
* **Resulting Pixel (after modification)**: 10011010 (binary)

By altering many pixels in this way, you can encode a full message or image in the LSBs, which is imperceptible to the human eye when looking at the image.

1. Explain the use of gray level slicing

**Gray Level Slicing** is a technique in image processing where you emphasize or highlight specific ranges of pixel intensity values (gray levels) while leaving the other values unchanged. This is often used for image enhancement or for highlighting certain features in an image.

#### Uses of Gray Level Slicing:

1. **Enhancing Specific Features**:
   * It can be used to isolate specific intensity levels (such as highlighting certain objects or regions) in an image by setting the pixel values within a given range to a fixed value (typically 255 or 0), making these regions more distinct.
   * For example, if an object in the image has a gray level in the range of 100-150, gray level slicing can be used to emphasize these gray levels and set them to 255 (white), making the object stand out from the rest of the image.
2. **Thresholding**:
   * Gray level slicing is closely related to thresholding. In thresholding, you convert pixel values that are above or below a certain threshold to black or white. Gray level slicing can be seen as a more flexible thresholding technique that works within a specific range of gray levels.
3. **Highlighting Textures or Patterns**:
   * In certain cases, different objects in an image have distinct textural or tonal patterns. By performing gray level slicing, you can highlight the tonal or textural differences between these regions and focus on those areas for further analysis, such as object detection or pattern recognition.

#### Types of Gray Level Slicing:

* **With Background**:
  + The pixels within a specific gray level range are set to 255 (white), and all other pixels are set to 0 (black). This highlights the selected region of interest while discarding all other information.
* Example:
  + For an image with pixel values in the range [100, 150], you might set all pixels in this range to 255 (white) and set all other pixel values to 0 (black), essentially isolating the target region.
* **Without Background**:
  + Pixels within the specified gray level range are set to 255 (white), and the rest of the image is left unchanged. This highlights the region but preserves the rest of the image’s details.
* Example:
  + For the same range [100, 150], you might set those pixels to 255 but leave all other pixels intact, making the highlighted region stand out but without losing the context of the original image.

#### Applications of Gray Level Slicing:

* **Medical Imaging**: Highlighting certain tissue types or abnormalities.
* **Object Detection**: Isolating certain objects in an image based on their gray level intensities.
* **Quality Inspection**: Identifying defects in manufacturing processes based on pixel intensities corresponding to defects.