

DIGITAL IMAGE PROCESSING - LAB 1

Introduction to MATLAB and Image Representation

OBJECTIVES:

1. Understand the MATLAB environment and basic commands.
2. Learn fundamental Digital Image Processing (DIP) concepts.
3. Explore different types of images: Grayscale, Binary, RGB, and Indexed.
4. Perform basic image conversions in MATLAB.

INTRODUCTION TO MATLAB

1.1 What is MATLAB?

MATLAB (Matrix Laboratory) is a high-level programming language and environment used for numerical computing, visualization, and application development. It is widely used in engineering, data science, and image processing.

1.2 MATLAB Environment Overview

When MATLAB starts, the following key components appear:

- **Command Window** – Where commands are entered and executed.
- **Workspace** – Displays variables created during the session.
- **Command History** – Stores a history of executed commands.
- **Editor** – Used for writing and running scripts.

1.3 Basic MATLAB Commands

Try the following in the **Command Window**:

```
clc;           % Clears the Command Window
clear;         % Clears all variables from the workspace
who;           % Displays all variables in the workspace
whos;          % Displays detailed information on variables
help plot;     % Displays help for a function (e.g., plot)
```

1.4 Creating and Manipulating Matrices

```
A = [1 2 3; 4 5 6; 7 8 9]; % Define a 3x3 matrix
B = A';                 % Transpose of A
C = inv(A);              % Inverse of A
D = A .* 2;              % Element-wise multiplication
```

INTRODUCTION TO DIGITAL IMAGE PROCESSING (DIP)

2.1 What is Digital Image Processing?

Digital Image Processing involves the manipulation of images using computational techniques. It includes:

- Image enhancement
- Image restoration
- Image segmentation
- Feature extraction

2.2 Image Representation in MATLAB

Images in MATLAB are stored as matrices where:

- **Grayscale Image:** 2D matrix with intensity values (0-255 for uint8 format).
- **Binary Image:** 2D matrix with only 0s (black) and 1s (white).
- **RGB Image:** 3D matrix with Red, Green, and Blue channels.
- **Indexed Image:** Uses a colormap to map intensity values to colors.

2.3 Reading and Displaying Images

```
img = imread('peppers.png');    % Read an image
imshow(img);                    % Display the image
```

2.4 Checking Image Properties

```
size(img)    % Returns the dimensions of the image
whos img     % Displays detailed information about the image
```

IMAGE CONVERSIONS AND BASIC OPERATIONS

3.1 Converting RGB to Grayscale

```
gray_img = rgb2gray(img);
imshow(gray_img);
```

3.2 Converting Grayscale to Binary

```
binary_img = imbinarize(gray_img, 0.5); % Convert to binary with threshold 0.5
imshow(binary_img);
```

3.3 Extracting Individual Color Channels

```
red_channel = img(:,:,1);  
green_channel = img(:,:,2);  
blue_channel = img(:,:,3);  
imshow(red_channel);
```

3.4 Resizing and Rotating an Image

```
resized_img = imresize(img, [200, 200]);  
rotated_img = imrotate(img, 45);  
imshow(rotated_img);
```

3.5 Saving an Image

```
imwrite(gray_img, 'grayscale_image.png');
```

LAB TASKS:

1. **Basic MATLAB Operations:** Perform matrix operations (addition, multiplication, transpose) in MATLAB.
2. **Read an Image:** Load the 'peppers.png' image and display its properties.
3. **Convert and Save:** Convert an RGB image to grayscale and binary, then save both.
4. **Extract Color Channels:** Extract and display the red, green, and blue channels separately.
5. **Resize and Rotate:** Resize an image to 200x200 pixels and rotate it by 30 degrees.