PAMANTASAN NG LUNGSOD NG MAYNILA
(University of the City of Manila)
Intramuros, Manila

Elective 3

Laboratory Activity No. 1

Image Acquisition and Manipulation

Score

Submitted by:

Bolocon, Joniel R.

De Guzman, Jastine V.

De Juan, Lord Welchie P.

Mercado, Ed-Vir G.

Puno, Harold Dennis R.

Saturday 7:00 AM – 4:00 PM / CPE 0332.1-1

Date Submitted

26-07-2024

Submitted to:
Engr. Maria Rizette H. Sayo

I. Objectives

This laboratory activity aims to implement the principles and techniques of image acquisition through MATLAB/Octave and open CV using Python

- Acquire the image.
- Rotate the image by 30 degrees.
- Flip the image horizontally.

II. Methods

A. CODES

a. Octave Code

```
pkg load image;  % Load the image package


% Read the image (use a simpler path for testing)
img = imread('flower.jpg'); %img must be in the current file path to connect file


% Rotate by 30 degrees
rotated_img = imrotate(img, 30);


% Flip horizontally
flipped_img = fliplr(rotated_img);


% Display results
figure(1);
imshow(img);
title('Original Image');
figure(2);
imshow(rotated_img);
title('Rotated 30°');
figure(3);
```

imshow(flipped_img);

title('Rotated & Flipped');

b. MATLAB Code

```
% Read the image
img = imread('flower.jpg');
% Rotate by 30 degrees
rotated_img = imrotate(img, 30);

% Flip horizontally
flipped_img = fliplr(rotated_img);

% Display results figure(1);
plot(1,1);
imshow(img);
title('Original Image');
figure(2);
plot(1,1);
imshow(rotated_img);
title('Rotated 30°'); figure(3); plot(1,1); imshow(flipped_img);
title('Rotated & Flipped');
```

B. Supplementary Activity (Python Code)

- Write a Python program that will implement the output in Method A.

```
import cv2
# Read the image
img = cv2.imread('flower.jpg') #img must be in the same folder as the Python program to connect file

# Rotate the image by 30 degrees
(h, w) = img.shape[:2]
center = (w // 2, h // 2)
rotation_matrix = cv2.getRotationMatrix2D(center, 30, 1.0)  # 30 degrees, scale = 1.0
rotate_img = cv2.warpAffine(img, rotation_matrix, (w, h))

# Flip the rotated image horizontally
flipped_horizontal = cv2.flip(rotate_img, 1)  # 1 for horizontal flip
```

```
# Display the images
cv2.imshow('Original Image', img)
cv2.imshow('Rotated by 30 Degrees', rotate_img)
cv2.imshow('Flipped Horizontally', flipped_horizontal)

# Wait for a key press and close all windows
cv2.waitKey(0)
cv2.destroyAllWindows()
```


C. Results

1. Octave Output

   picture file: flower.jpg

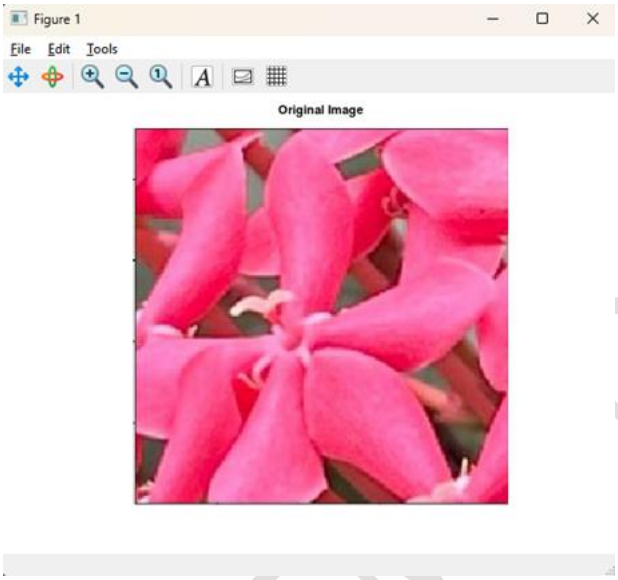

Figure 1: Original Image of Octave

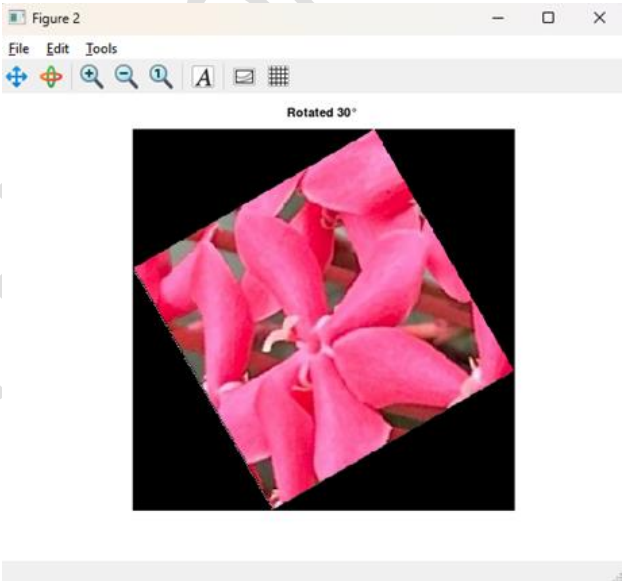

Figure 2: Rotated by 30 degrees of Octave

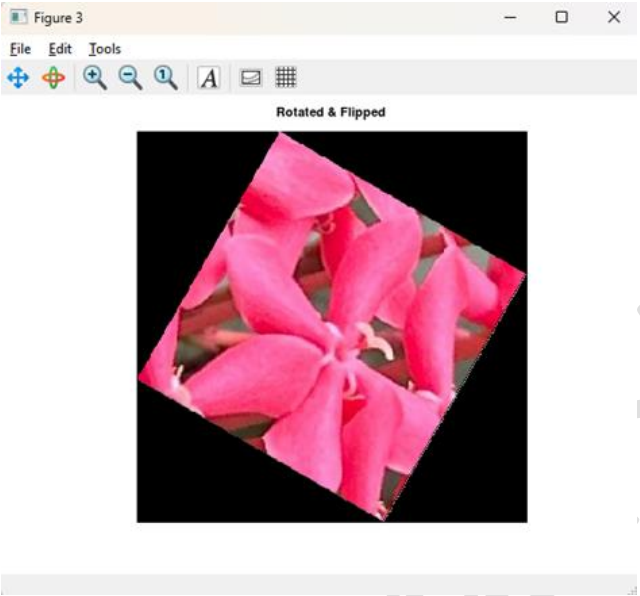

Figure 3: Flipped horizontally of Octave

2. MATLAB Output

picture file: flower.jpg

Figure 4: Original Image of MATLAB

Figure 5: Rotated by 30 degrees in MATLAB

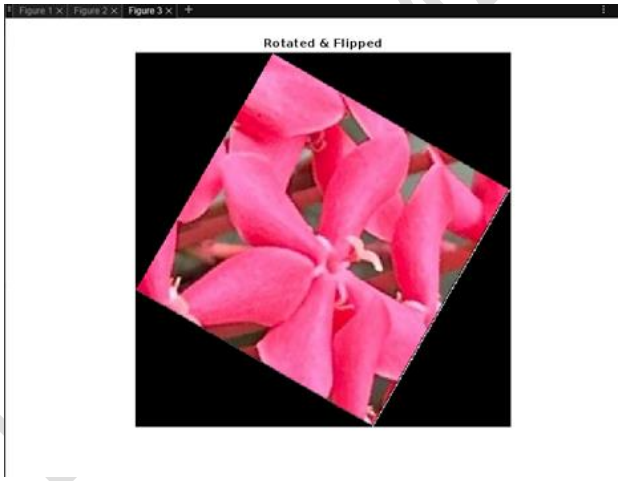

Figure 6: Flipped horizontally of MATLAB

3. Python Output

Figure 7: Original Image in Python

Figure 8: Rotated by 30 degrees in Python

Figure 9: Flipped Horizontally in Python

4. Visualize the results, analyze and interpret:

The output in each software was different, for example in MATLAB, it displayed it differently as it has a different frame, and it is centered differently compared to Octave where it is shown with a better frame and centered correctly. The Python code created displayed the image itself while having no frame and the text above it, however, the flipped horizontal seems to be too small for the preset frame and it needs to be maximized to be shown fully. Each software probably has its unique way of outputting the code such as using packages in Octave. As observed, MATLAB has built-in packages that allow ease of use, while Octave and python both need packages to be installed separately such as image and cv2. Nonetheless, the desired output has been met and the method of outputting is up to the technique and software preference of the coder in line with the requirements.

III. Conclusion

After successfully performing this laboratory exercise, we therefore conclude that it gave us an understanding of the basic methods of image display and manipulation using MATLAB/Octave and Python, which specifically are acquiring, rotating, and flipping an image horizontally. Moreover, using the packages for Octave and Python, we have comprehended the similarities of MATLAB and Octave on syntaxes, transpiring as well on code structure format. However, implementing the method in a python program provided a cropped version of the output, proving that either MATLAB or Octave must be utilized in terms of digital image processing. Overall, this laboratory exercise has been successful with MATLAB, Octave and OpenCV on P.

References

[1] D.J.D. Sayo. “University of the City of Manila Computer Engineering Department Honor Code,” PLM-CpE Departmental Policies, 2020.