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:

Group 1

Antoni, Austine C.

Atencia, Dan Eric B.

Bauyon, Jared Randalle B.

Jao, Steven Rey C.

Leonardo, Aibel Renzo T.

Saturday, 7AM-4PM / CPE 0332.1-1

Date Submitted

25-07-2024

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. Perform a task given in the presentation

- Copy and paste your MATLAB code

```
% Read the image
img = imread('E:\PLM CET SUBJECTS\Digital Image Processing\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

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

```
import cv2

#Acquire the image
img = cv2.imread('flower.jpg')

#Rotate the image by 30 degrees
#Get image dimensions
(h, w) = img.shape[:2]
(centerX, centerY) = (w // 2, h // 2)

#Define the rotation matrix
angle = 30  #angle in degrees
scale = 1.0  #scale factor
rotation_matrix = cv2.getRotationMatrix2D((centerX, centerY), angle, scale)

#This function rotates the image by 30 degrees
rotated_image = cv2.warpAffine(img, rotation_matrix, (w, h))

#Flip the image horizontally
flipped_img = cv2.flip(rotated_image, 1)

#Display images
cv2.imshow('Original Image', img)
cv2.imshow('Rotated Image', rotated_image)
cv2.imshow('Rotated & Flipped Image', flipped_img)
#Wait until user presses a key
cv2.waitKey(0)
```

C. Results

a. MATLAB results:

Figure 1: Acquire an Image of a Flower

Figure 2: Rotate by 30 degrees

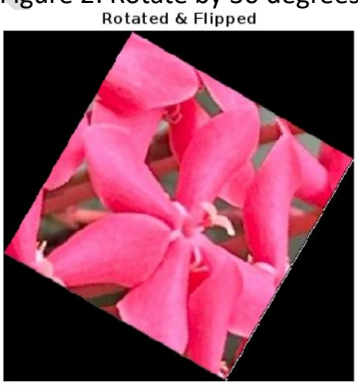

Figure 3: Flip horizontally

b. Octave results

Figure 1: Acquire an Image of a Flower

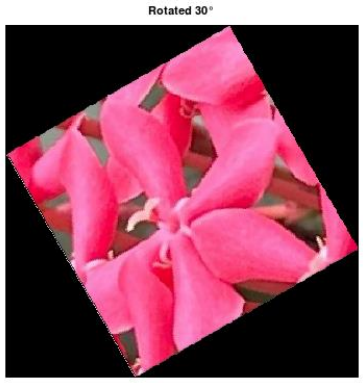

Figure 2: Rotate by 30 degrees

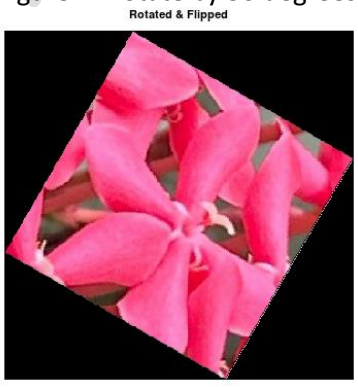

Figure 3: Flip horizontally

c. OpenCV results

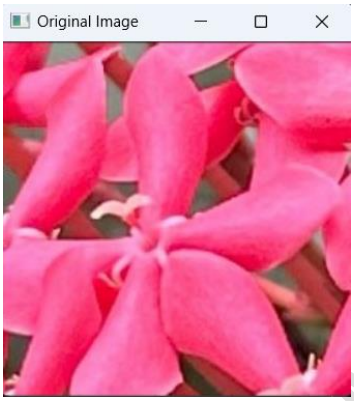

Figure 1: Acquire an Image of a Flower

Figure 2: Rotate by 30 degrees

Figure 3: Flip horizontally

Visualize the results, analyze and interpret:

MATLAB/Octave Explanation:

In Octave, in order to manipulate and acquire an image, “pkg load image” is loaded first. For both Octave and MATLAB, to read an image, function “imread(‘file path’)” is used to read an image of a specified path. First, to rotate an image, “imrotate(variable, angle)” is used. If the user wants to rotate an image by an angle, for example 30 degrees, the syntax should be: “imrotate(img, 30);” Second, to flip an image horizontally, “fliplr” is used. It rotates the image from left to right, thus, “lr”. In the program, we are flipping the rotated image, so, the syntax should be “fliplr(rotated_img);”. Lastly, to display an image, a new figure is created: “figure(1)”. The function “imshow(variable_name)” displays the image according to variable name. The function “title()” sets the title of the window of a figure.

OpenCV Explanation:

First, to use OpenCV and load its functions, “import cv2” will import then OpenCV library. Similar to MATLAB, “imread(‘path’)” is used to read an image located at the defined file path. To rotate an image using OpenCV, a set of functions are used. Unlike MATLAB, only one function is used to rotate an image. In OpenCV, first, the dimensions of the image are used and retrieved by the function “var_name.shape()”. In this program, we are only using the height and width, thus, we slice it by [:2]. Then, to properly scale the images, the center coordinates are calculated by dividing (or flooring) the 2 dimensions by 2. Now, we are defining the rotation matrix by setting the rotation angle and image scale. The function “cv2.RotationMatrix2D()” rotates an image by angle around the center point with a specified scale. Lastly, to apply the rotation matrix to the image, “cv2.warpAffine()” function is used. The parameters are: (img_var_name, rotation_matrix,(w,h)). Moreover, to flip an image, “cv2.flip()” is used. The parameters are: (var_name, 0 or 1) “1” is the value for flipping an image horizontally, whereas, “0” is the value on a vertical flip. Next, to display an image, “cv2.imshow()” is the function used. The parameters are: (‘window_name’, var_name). Lastly, to display an image indefinitely, “cv2.waitKey()” is used. The argument ‘0’ enables an infinite wait, this means that the function will wait until a key is pressed.

IV. Conclusion

In this laboratory activity, we learned to acquire, process, and manipulate an image through MATLAB/Octave and OpenCV using Python. Specifically, in this activity, we acquired, rotated, and flipped an image using MATLAB, Octave, and OpenCV. The three platforms are mostly similar but differ in declaration and implementation. We acquired an image using the *imread()* function for all platforms. MATLAB and Octave have a straightforward function to rotate an image by angle degrees. However, OpenCV doesn't. It will need to define first the rotation matrix, then apply it by *warpAffine()*. Lastly, all platforms have a straightforward function to flip an image horizontally. MATLAB and Octave uses *fliplr(),* while OpenCV uses *flip(var_name, 0 or 1).*

References

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