

# HW3, 10 pts



**The Hashemite University**  
**Computer Engineering Department**  
**Computer Vision**  
**Fall 2022**

**Dr. Khalil Yousef**

---

## **Lab Instructions:**

- This lab assignment must be turned electronically using the Moodle submission Wizard.
- This is an individual lab -- each student should submit their own lab.
- In every uploaded file, include your name, date, and the assignment name.
- Untested code is buggy. Your code should compile and run when I test it (or you demo it). If it does not compile & run, you will get zero (No tolerance).
- Use PYTHON (Jupyter Notebook) to do your lab assignment.
- No makeup for this assignment is allowed.
- Please start early and ask me for help if you get stuck.

# Assignment 3: Homographic Image Warping

## TASKS


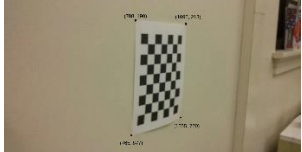

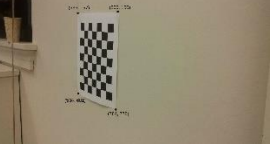
- Use your webcam to capture an image of your face similar to what you have done in Assignment1.
- Use the suitable python command(s) to just crop the image of your face from the background.
- Compute the homography between the cropped face image and each of the three given images. This is done as was discussed in Lec2 and end of Lec4 of the course by forming the  $A_{8 \times 9}$  matrix and computing the null space, which would be the solution (i.e. the intended homography in a vector shape). You then need to use the reshape command to convert the homography (vector shape 9 by 1) to the matrix shape 3 by 3. Make sure to divide the computed homography by the very last element (h33) so that the final value of h33 of each homography is one.
- Project your cropped face image into the 3 given images (Exactly on the identified chess board calibration pattern, which coordinates are given) using the computed homographies. You will need to create a nested for loop (i.e. two for loops) that would go over all the rows and columns of your face image, apply the computed homography to each pair of these coordinates (row and column), find the physical coordinates of the computed result in the given input image, and finally copy the associated color value from your face image from all the three color channels to these the physical coordinates.

## Deliverables


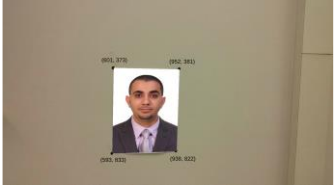

- A single python file (\*.ipynb) divided into cells, following the assignment's tasks above and template below.
- Your cropped face image.
- The resulted images that show your projected face.

## Expected results:

Inputs:

Face Image	Img1	Img2	Img3
			

Outputs:

Img1	Img2	Img3
		

# Assignment 3.m

# Name: \_\_\_\_\_

# Date: \_\_\_\_\_

# Semester: Fall 2022

# Computer Vision Course

#

# This is the template for your Assignment 3 script.

# It is not intended to be run straight through from beginning to end.

#

# Instead, you will execute each "cell" as you work through the assignment,

# and I can execute individual cells as I look at your work.

#

# You can execute a "cell" by typing Ctrl-Enter with the cursor in that cell.

#

#

# PLEASE KEEP THE NAMES AND NUMBERS OF THE CELLS THAT ARE HERE THE SAME.

# But you can add additional cells if you like.

#

## 1. Capturing and recording/writing an image using your webcam.

## 2. Cropping your face image, display it, and saving it using the suitable command

## 3. Reading all images (your face image and the three given images)

# Img\_face =

# Img1 =

# Img2 =

# img3 =

## 4. Computing and printing the homography that transforms Img\_face to Img1

# H1 =??

## 5. Computing and printing the homography that transforms Img\_face to Img2

# H2 =??

```
## 6. Computing and printing the homography that transforms Img_face to Img3
# H3 =??
```

```
## 7. Projecting Img_face to Img1 using H1 and display the resulting image img1
# show and save Img1
```

```
## 8. Projecting Img_face to Img2 using H2 and display the resulting image img2
# show and save Img2
```

```
## 9. Projecting Img_face to Img3 using H3 and display the resulting image img3
# show and save Img3
```