

ENPM 673: Perception for Autonomous Robots
Midterm Exam 2022
3/15/2022

Due on 3/16/2022 at 7 pm

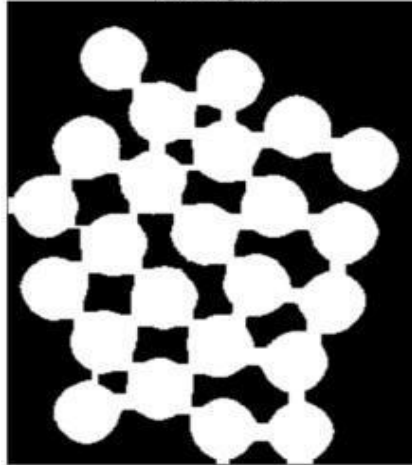
Instructions (please read):

- **Keep the answers as short** as they need to be, don't make them unnecessarily verbose.
- When asked about the steps that lead to performing a certain task or the image processing pipeline, just list the name of the algorithm that needs to be used at each step and why.
- You are **allowed to use inbuilt functions** for the steps involved in your pipeline. However, you cannot use functions or APIs that directly solve the problem. For example, in questions 2 and 3, you **cannot use APIs** like `cv2.calibrateCamera` or `cv2.createStitcher` since they can directly solve the problem.
- Your submission on ELMS/Canvas must be a zip file, following the naming convention **YourDirectoryID_midterm.zip**. If your email ID is `abc@umd.edu` or `abc@terpmail.umd.edu`, then your Directory ID is `abc`. Remember, this is your directory ID and NOT your UID.
- Please **submit the python script(s)** you used to compute the results, along with one **.pdf file** with solutions for all the questions and a detailed **README.md** file to run your code. The `.py` scripts must be named as *question1.py*, *question2.py* etc and the report must be named by **YourDirectoryID.pdf**.
- Refer the drive link for the images, don't crop the thumbnails to use it as inputs.
- There are four questions to be solved.
- Please follow the directory structure as follows:

```
|— YourDirectoryID_hw1.zip
    |— Code
        |— .py files
        |— any subdirectories that you may have
    |— YourDirectoryID.pdf
    |— README.md
```

Question 1 (20 marks): Assuming the image below represents an x-ray of old coins. Read [this image](#) seen below using OpenCV taking into consideration that this is a binary image.

1. Write a program that will separate each coin from each other, so that the output image has the coins completely separated.
2. Write a program that will automatically count how many coins are there in the image (from step 1) . Note: You are allowed to use **any** built-in function in OpenCV. However, you are required to explain each step you took in your solution (Use pipeline or a block diagram).

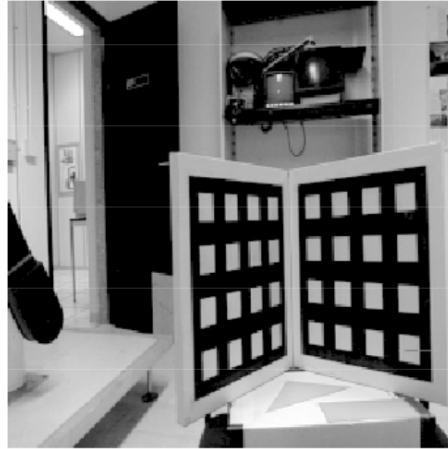


Question 2: (30 marks) Given two images, imageA and imageB [in this link](#) shown below, find the matching points between these two images and stitch them together by finding the homography between them. Write the pipeline that you used to solve this problem and attach the code solution.

Note: You are not allowed to use any API that does stitching directly, such as (OpenCV `createStitcher.stitch()`). Inbuilt functions are allowed to implement the intermediate steps involved.



Question 3 (30) Calibrate the camera (Find the intrinsic matrix K).



1. What is the minimum number matching points to solve this mathematically?
2. What is the pipeline or the block diagram that needs to be done in order to calibrate this camera given the image above.
3. First write down the mathematical formation for your answer including steps that needs to be done to find the intrinsic matrix K
4. Write a program that will calibrate the camera given the point correspondences from world to image. Use the data provided below. Please note you are only allowed to use numpy for this question. No marks will be given if you use any other library/tool.

Image points		World points		
x	y	X	Y	Z
757	213	0	0	0
758	415	0	3	0
758	686	0	7	0
759	966	0	11	0
1190	172	7	1	0
329	1041	0	11	7
1204	850	7	9	0
340	159	0	1	7

Question 4: (20 marks) Implement K-means algorithm to separate this [image](#) below, based on color, into 4 classes. Note: You are NOT allowed to use any built-in function, implement your code from scratch. Explain each step you take.

