Practicum #4

Date Given: July 04, 2022
Due Date: July 08, 2022 at 11:59 pm

CSCI 4261 - Introduction to Computer Vision
Faculty of Computer Science, Dalhousie University

Plagiarism Policy

- This practicum is an individual task. Collaboration of any type amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Content should not be copied from any source(s). Please understand the concept and write answers in your own words.
- If you wish to learn more Dalhousie Academic Integrity policy, please visit the following link: https://www.dal.ca/dept/university_secretariat/academic-integrity.html

Assessment Criteria

Task Assessment:

Each question will be marked on the following criteria:

- 100% - 90% marks:
  The solution you have provided is correct and match all the expected requirements.
- 90% - 80% marks:
  The solution is correct but there are some areas of improvement or context missing.
- 80% - 70% marks:
  The solution is close to the correct answer but your approach is correct.
- 70% or less marks:
  Your solution is not correct and there are obvious loops in your understanding.

Requirements:

Implement a simple registration algorithm from scratch.

You will do 3 task that will help you to reach the goal. Markers will test your code with a different image, so double check that your code runs smoothly (Running time will be a factor). In your repository, you must provide 3 functions:

- warping_affine([Sx Rx Tx;Ry Sy Ty;0 0 1],Image) for Task -1
- align_2dof(input,reference) for Task -2
- affine_registration(input,reference) for Task -3

IMPORTANT NOTE:

- For Task-1 and Task-2 the only libraries that you need are numpy and matplotlib **anything else is not allowed**.
- For Task- 3 you can use any library

Task - 1

Implement a warping function using an affine matrix T. Then use it to transform img1. Try different transformations and check that your results are as expected.

Include in your PDF report

1. A figure that contains the original image, and one scaling (50%), one translation (50 pixels in x and 100 in y) and one rotation (30 degrees).

Task – 2

Implement your alignment function. Use img2 as reference and transform img1 to be aligned with im2. In this task you can use brute force (you are free to use any other strategy as long it is automatic) to test the change in position until you find the transformation T that maximize or minimize your similarity metric. You can use any metric of similarity.

Write in your PDF report

1. What was your approach for the search?
2. Why did you select the similarity metric?
3. The optimal matrix T.

The img2 is just translated from img1, so focus your search in only changing the Tx and Ty values of the affine matrix. A maximum of 10 minutes is allowed to find the solution (In the Marker's computer), if your function does not find the optimal T in that time or less, you won't get the points.

Task - 3

Test your creativity, improve your algorithm to register im1 and img3. Use img3 as reference and modify img1(input). In this case you will need to change scaling, rotation and translation to find the optimal T. Modify your search strategy. Looking for each single combination will take long time. A maximum of 30 minutes is allowed to find the solution (In the Marker's computer), if your function does not find the optimal T in that time or less, you won't get the points.

Write in your PDF report

1. The strategy that you used to minimize the running time
2. The optimal T.

Make Your Life Easy [TIPS]

Depending on the transformation, warping can result on different image size. For task 2 and 3, keeping the size of the new image equal to the target image will reduce the complexity of your code and will make it easy to measure the similarity. In terms of time, remember the trick of the pyramid of gaussians used in SIFT that uses different scales, this can help you to find the right position faster. I know that you don't know the hardware that the Marker will use to test your code. So, keep a buffer of time just in case. In real life we don't know what the consumer will use to run our implementations and the expectations is that the code runs efficiently.

Submission Criteria

The submission for this Practicum will be done on 2 platforms:

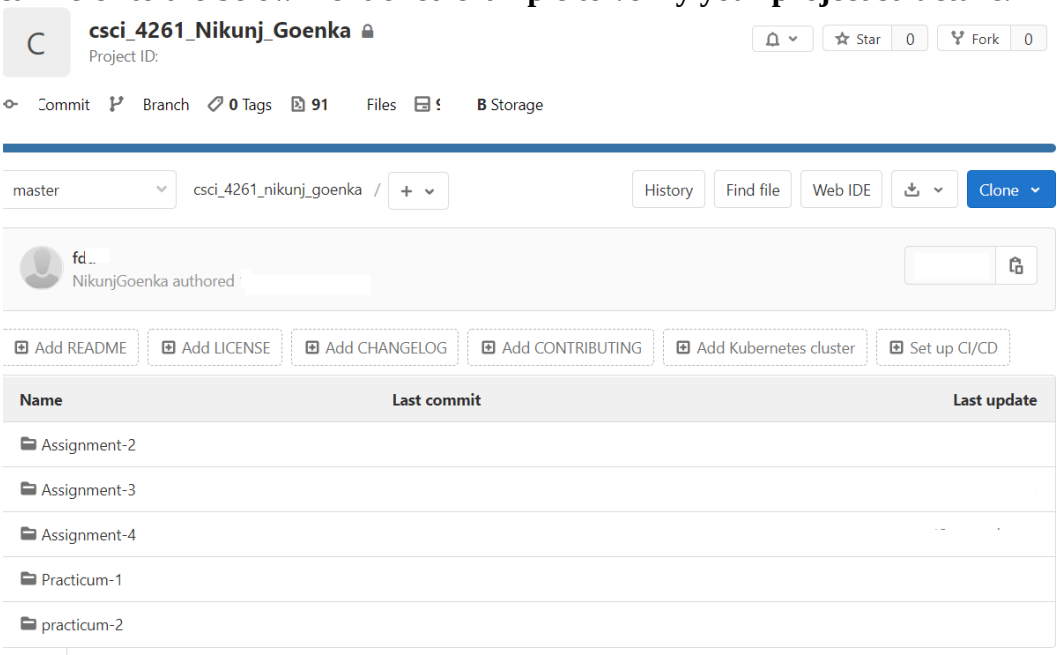
1. **Document Submission:**
   The documentation should be a PDF that contain the following:
   a. Answers to the questions asked.
   b. Mention the link to your GitLab repository.
   Submit the PDF on Brightspace before the practicum deadline.
   NOTE: The answers should be clearly marked and in a proper format.

2. **Code Submission:**
   For the code submission follow the mentioned steps.

   Step1: Create a project named "CSCI_4261_FName_LName" on GitLab. (https://git.cs.dal.ca/)
   Step2: Add a new directory named "Practicum_4".
   Step3: Add your '.py' or '.ipynb' files containing the code required for the practicum.
   Step4: Add **"@vjaiswal"** as the maintainer to the repository.

   Alternatively:
   You can also submit your code on GitHub (https://github.com/). Add "**GoenkaNikunj97**" to your GitHub repository.

You can refer to the below mentioned example to verify your project structure.



Upload you code before the practicum deadline, code pushed after the deadline will not be marked. You can name you code files according to your preference but the name should clearly indicate the task and subtask they are associated with.

Failure to follow the submission criteria can result into 10% deduction in marks.