Middle East Technical University Department of Computer Engineering

CENG 483

Introduction to Computer Vision

Fall 2022-2023

Take Home Exam 2

Corner Detection with Harris Corner Detector

Due date: **11 December 2022, 23:55**

1 Objectives

The purpose of this take home exam is for you get hands-on experience with key point detection by using Harris Corner Detector.

2 Background

In this assignment you are required to implement different versions of Harris Corner detector and visualize the obtained keypoints in given images. The implementation will be divided into different tasks and the code and discussions should be provided as a jupyter-notebook for each individual task.

The text below continues with detailed explanations of the methods and requirements.

3 Harris Corner Detector

Harris Corner Detector finds corners/keypoints based on the change of intensity in an image patch after shifting that image patch in 8 different directions. Corners are such regions so that any kind of shift in any direction will result in relatively big change of intensity. In this assignment, you will implement different versions of Harris Corner Detector as explained in lectures which will output images together with top 10 corners that obtain the best score.

3.1 Task 1: Naive Formula + Uniform Weighting

In this task, you will implement the formula given in 24. lecture slide together with uniform weighting of $w(x, y)$.

3.2 Task 2: Naive Formula + Uniform Weighting + Non-Maximum Suppression

In this task, you will implement non-maximum suppression on top of your implementation in task1.

3.3 Task 3: Taylor's Approximation + Uniform Weighting + Non-Maximum Suppression

In this task, you will implement the formula given in 29. lecture slide together with uniform weighting of $w(x, y)$.

3.4 Task 4: Smaller Eigenvalue as corner score + Uniform Weighting + Non-Maximum Suppression

In this task, you will use smaller eigenvalue as corner score (slide 40) together with uniform weighting and non-maximum suppression.

3.5 Task 5: R function as corner score + Uniform Weighting + Non-Maximum Suppression

In this task, you will use R function, given in slide 41, as corner score together with uniform weighting and non-maximum suppression.

3.6 Task 6: R function with fast windowing based on filtering + Uniform Weighting + Non-Maximum Suppression

In this task, you will perform fast implementation, given in slide 51-52, option 1 and non-maximum suppression.

3.7 Task 7: R function with fast windowing based on filtering + Gaussian Weighting + Non-Maximum Suppression

In this task, you will perform fast implementation, given in slide 51-52, option 2 and non-maximum suppression.

4 Implementation and Report

You need to tune your corner detector such that you will only show top 10 corners. In order to show top 10 corners first apply non-maximum suppression (if task requires you to include NMS) then filter the top 10 corners. You will be given 3 images with their rotated versions and report corners you obtain after each task. Additionally calculate average time to calculate corners for each image for each task and include it in jupyter-notebook. Also include a brief discussion in between tasks to discuss pros and cons of each approach.

5 Restrictions and Tips

- Your implementation should be in Python 3.
- You should use numpy. However, you can use other tools to convert images into numpy arrays and to apply convolution (e.g. scipy convolve2d). But in the rest of the implementation you should use numpy.
- Implementation of the corner detector must be your own, usage of functions such as cv2.cornerHarris is **forbidden**.

- Do not use any available Python repository files without referring to them in your report.

- Don't forget that the code you are going to submit will also be subject to manual inspection.

- Coming up with a correct implementation is important. We may randomly check for random configurations.

6 Submission

- **Late Submission:** As in the syllabus.

- Both the code and discussions should be in the given jupyter-notebook which should be uploaded on ODTÜCLASS before the specified deadline as a compressed archive file whose name is **<student_id>_the2.tar.gz**, e.g., 1234567_the2.tar.gz.

7 Regulations

1. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations.

2. **Newsgroup:** You must follow the course web page and ODTÜCLASS (odtuclass.metu.edu.tr) for discussions and possible updates on a daily basis.