UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Engineering, Built Environment and IT
Department of Computer Science

COS791

Image Analysis and Understanding

Due: 7 September 2024

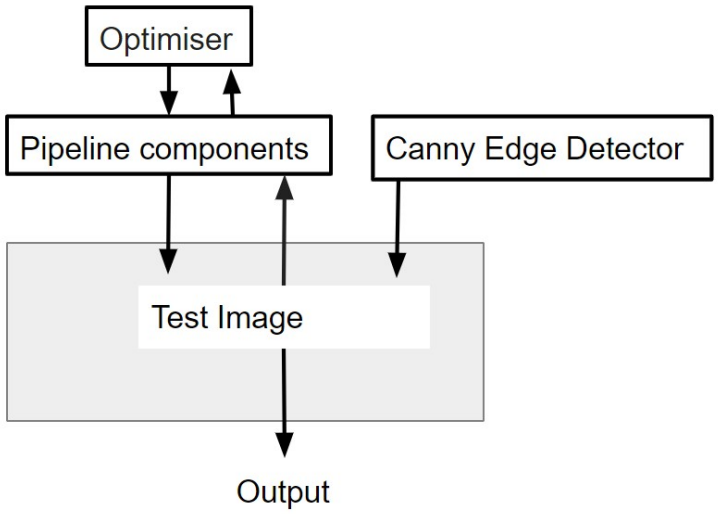
Instructions

1. This is an individual assignment.
2. Plagiarism is not allowed.
3. The use of libraries is permitted, you may use any programming language of your choice.
4. The assignment consists of one question which is to be answered as follows. The code, input-output image files, and a readme file specifying how to run your code should be placed in a folder labelled Ass1 zipped and uploaded. Additionally, a report (PDF) detailing the configuration and experimental setup and the results should be submitted.
5. Assignment image files are provided. There are 11 images.
6. This assignment is worth 25 Marks (Report is 10 Marks)

****

1. An edge detection pipeline involves the following steps.
    1. **Noise Reduction**
        (a) Objective: To mitigate the impact of noise, which can interfere with edge detection.
        (b) Options: Gaussian blurring, median filter, a bilateral filter etc.
    2. **Gradient Calculation**
        (a) Objective: To determine the rate of change of pixel intensity in both horizontal and vertical directions.
        (b) Options: Sobel, Prewitt, or Laplacian operators are often employed.
    3. **Edge Thinning (Non-Maximum Suppression)**
        (a) Objective: To refine the detected edges to single pixel width.
        (b) Options: By comparing the gradient magnitude of a pixel with its neighbours, pixels that are not local maxima in the gradient direction are suppressed.
    4. **Thresholding**
        (a) Objective: To differentiate between edges and noise.
        (b) Options: Hysteresis thresholding, binary thresholding.
    5. **Edge Linking**
        (a) Objective: To connect discontinuous edge segments.
        (b) Options: Hough transform, gradient direction analysis, region growing etc.

2.



In this assignment, you are expected to

1. Develop and apply a simple iterative optimiser which creates an edge detection pipeline from edge detection components. The pipeline components are elements defined from the steps stipulated in listing 1-5 which you have to parameterise. NB the listing is not exhaustive.
2. Two pipelines have to be created a) a general pipeline developed on one image and then applied to all. b) Each image will have its own pipeline.
3. The effectiveness of the pipeline is compared to the standard Canny Edge Detector's (which you have to develop using default values or a library can be used) output through the reference image.
4. The following metrics are to be used and reported on, feature descriptor (ORB/SIFT), mean-square-error(MSE), peak-signal-noise-ratio, and structural-similarity.
5. The results are to be tabulated and presented in the report. The report must also contain output images, and details of the optimiser including the objective function with justification. The best pipeline configuration for each image must also be reported. A critical analysis section should also be included.

1. Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), pp.679-698.