BLG 453E Homework - 3

Due 18.11.2017 23:59

Policy: Please do your homework on your own (Do not copy paste your solutions from the internet or your friends). The code and the report you submitted must be your own work. All code must be implemented using **Python** programming language and **OpenCV Python wrapper**.

For your questions: albay@itu.edu.tr

1. **Canny Edge Detection**

   (a) Implement the simplified version of the Canny edge detector. The syntax of your function should be as follows: $[E, M, A] = canny(I, sig, tau)$, where E contains the detected edges, M contains the smoothed gradient magnitude, A contains the gradient angle, I is the input image, sig represents σ for the smoothing filter, and tau is the threshold τ. You do not need to implement hysteresis thresholding, but you do need to implement oriented nonmaximal suppression.

   (b) Run your edge detector on Fig2wirebond_mask.jpg using $\tau = 5$ and the following three values for $\sigma^2 : 0.5, 1$ and 3. Discuss how the choice effects the results.

   (c) Apply your edge detector to Fig3bottles.jpg adjusting τ and σ as you see fit (Here by fitting we mean extracting clean boundaries of the bottle as well as level of liquid inside the bottle, while suppressing all other edges). Display the resulting edges and the parameter settings used.

   (d) Apply image gradient operator to your image and calculate image gradient magnitude ($\sqrt{I_x^2 + I_y^2}$). Apply a threshold to your gradient magnitude image, display your result. Compare and discuss about the result with built-in Canny edge detector function.

   Related Article: J. Canny, "A computational approach to edge detection", in IEEE Transactions Pattern Analysis and Machine Intelligence (IEEE PAMI), 8(6), pp. 679-698, 1986. (read this only if you want more detail)

2. **Feature Matching using SIFT**

   (a) Find matching points using built-in SIFT function. Show matching points on images. Use ″cameraman1.jpg″ and ″cameraman2.jpg″ for finding features.

   (b) Estimate the 2D affine transformation A matrix parameters (using enough number of them - $a_{11}, a_{12}, a_{21}, a_{22}$) by using the corresponding features from both of the images.

   (c) Use the estimated A matrix to transform the original image. How well does it line up with ″Cameraman2.jpg″? Try to analyze your transformation results. What are the possible sources of error?