ECSE 683 – Topics in Computer Vision and Robotics
Assignment 1 – Image Features
September 26, 2016

Due: October 10, 2016 at 5:00 pm

In this assignment we will investigate the extraction of intensity and range image features using the approaches discussed in class. Images referred to in the questions below are contained in the folder along with this file on myCourses.

Question 1

The corner detection algorithm discussed in class is a good example of operator design – in this case abstracting SSD as a measure computed from partial derivatives of x and y, and then using the rank of the resulting quadratic form as an indicator for the presence of a corner. An obvious question is how sensitive is this algorithm to the particular estimators used to determine I_x and I_y.

Do the following: ran range of signal max range

smoothing and differencing

a) First we will need a ground truth reference. Using your favorite image-editing tool, e.g., GIMP, open the included file *checkerboard.jpg* and identify the pixel coordinates of at least 10 clearly identifiable corners. Prepare 3 additional versions of checkerboard.jpg by adding 5%, 10%, and 25% additive Gaussian noise. This will serve as the test set.
b) Implement, using Matlab or equivalent (e.g. Octave), the corner detection algorithm discussed in class. Here, build discrete operators to estimate the partial derivatives I_x and I_y. Your operators should also include some smoothing. Compare your error against ground truth for each of the 4 images in the test set.

gets a scaler value from the function you fill with the data

c) Repeat Part (b), but this time use least squares to obtain a parameterization of each local neighborhood using a suitable polynomial. The partial derivatives I_x and I_y now reduce to polynomial coefficients.
d) Compare the performance of the two implementations on a real image, *find_the_corners.jpg*.

Briefly summarize your observations, highlighting any major differences (if any) between the two approaches. Some experimentation will be required to determine the appropriate neighborhood sizes both for approximation of the partial derivatives and matrix C.

Question 2

Next we consider the detection and localization of intensity edges. Following the procedure outlined in CANNY EDGE DETECTOR (Section 4.2.2 Trucco+Verri), design an edge detector to do the following:

i) Run your edge detector on image lena.png. Show intermediate and final images: the norm of the image gradient, the image after non-maximum suppression, and the image after hysteresis thresholding. Start with (Start with $\sigma = 3$. Choose an appropriate filter size. Choose the optimal low and high threshold. Label your results with the parameters you used.)

ii) Evaluate your algorithm on edge location performance using synthetic images (snr1.png through snr100.png), and compare it to the built-in Matlab function. The number of edge pixels should be the same as the image height. Good detectors should produce the most edge pixels along the central line and the fewest in other areas. For each image, present side-by-side results generated by your code and those generated by Matlab command "edge(I,'canny',...)". Try and use parameters, which produce the best results. (The results being compared should use the same parameters.) Provide a brief discussion of your results.

Question 3

Finally, we consider features in range images. See file problem3_readme_data.txt in the assignment folder for information on the data to be used for this question. Do the following:

i) Begin by implementing operators for I_x, I_y, I_{xx}, I_{yy}, and I_{xy} using a 2D Gaussian as a basis. The optimal value of σ will have to be inferred from the data. With these in place, compute the H-K map for the angel data set. Display your results accordingly.
ii) Compute the shape label image, S, by assigning a shape label s_i to each pixel, according to the rules in Table 4.1 of Trucco+Verri. (Hint: you should take the practical considerations mentioned in the Section 4.4.2 into account).
iii) Now repeat Part (ii) using a least-squares approximation to estimate the values of a polynomial $h(x,y) = ax^2 + bxy + cy^2 + dx + ey + f$ at each coordinate. You need to figure out the appropriate sampling neighborhood. From the polynomial coefficients, determine H and K. Generate the H-K map for the angel data set. Display your results and compare against those obtained in Part (i).
iv) Repeat Part (ii) using the H-K map obtained using least squares. Compare against the shape label image you obtained initially.

fpf/September 26, 2016