CS-512 – Assignment 3 (5%)

Feature detection and matching

Due by: October 16, 2018

Review questions

Answer the following questions. Make sure that your answers are concise. In questions requiring explanations, make sure your explanations are brief.

1. Corner detection
   (a) Explain the basic principle of corner detection. How is the number of principal directions assessed?
   (b) Explain how PCA (Principal Component Analysis) is used to find principal directions of gradient orientations in a local patch.
   (c) Given the gradient vectors $\{(0,0),(0,1),(0,2),(0,3),(0,4),(1,0),(1,1),(1,2),(1,3)\}$ compute the correlation matrix that can be used for corner detection.
   (d) Write the condition on the eigenvalues of the gradient correlation matrix that is used for corner detection.
   (e) Explain how non-maximum suppression works for corner detection.
   (f) Explain how Harris corner detection avoids computing the eigenvalues of the gradient correlation matrix directly.
   (g) Write and explain the formula for computing better localization of a corner. What is the condition for the solution in the formula to exist?
   (h) Explain how feature points can be characterized using HOG (Histogram of Oriented Gradients). What are the requirements from a good characterization of feature points?
   (i) Explain how SIFT features are computed.
2. Line detection
   (a) Explain the problem of using the slope and y-intercept as line parameters when using the Hough transform.
   (b) Given a line with a slope of 45 degrees that passes at a distance of 10 from the origin, write the value of the coefficients a, b, c in the explicit line equation $a \cdot x + b \cdot y + c = 0$. To verify your answer, draw the line, find points on it, and see that they satisfy the explicit line equation.
   (c) When using the polar representation of lines, what does the vote of each point in the image looks like in the parameter plane?
   (d) Explain how are lines detected by checking the parameter plane.
   (e) Explain the trade-off regarding bin size in the parameter plane.

(f) How can voting in the parameter plane be improved if the normal at each voting point is known.

(g) When using Hough transform for circles, what is the number of dimensions of the parameter space?

3. Model fitting

(a) Explain the disadvantage of using the equation $y = a \cdot x + b$ for line fitting. What kind of lines cannot be fitted accurately using this equation?

(b) Given a line with a normal $(1, 2)$ and distance of 2 from the origin, write the value of the vector l representing this line in the explicit line equation $l^T x = 0$. Note that l is a 3×1 vector.

(c) Explain how to fit a line using the explicit line equation. Write the equation that has to be solved for the unknown line parameters.

(d) Given the points $\{(0, 1), (1, 3), (2, 6)\}$ write the 3×3 matrix that has to be formed to find the parameters of the line that fits the points.

(e) Write the explicit equations for conic curves. What is the constraint on the parameters a, b, c, d, e, f that guarantees that the model will be an ellipse?

(f) Write the equation that needs to be solved for fitting an ellipse using algebraic distance. Explain which points on the ellipse affect more the fitting (points close to the long axis or short axis of the ellipse).

(g) Write the objective function that needs to be minimized when fitting an ellipse using geometric distance. What is the additional complication involved when trying to determine the ellipse parameters by minimizing this function?

(h) Write the objective function for active contours, and explain the components in it.

(i) Explain how the continuity and curvature of a discrete curve can be estimated when using active contours.

(j) Explain how the continuity of active contours may be relaxed to allow for sharp corners.

Programming questions

In this assignment you need to implement ONE of several possible applications as described below. Note: you only need to implement ONE of the options.

1. When capturing images from a camera allow the program to capture and process the images continuously.

2. The input image should always be converted to grayscale before processing it.

3. The main parameters of each algorithm should be made available for interactive manipulation through keyboard/mouse/trackbar interaction.

4. You may NOT use the OpenCV function which directly implements the algorithm you choose. You may, however, use it for verification purposes. You may use other OpenCV functions as necessary.

5. Your program must include a help key describing its functionality.

6. You need to evaluate the performance of the algorithm you choose using test data (e.g. adding noise or occlusions). The results of your evaluation should be included in your report. Try to determine the strengths and weaknesses of the algorithm.

7. Your report must include a description of the algorithm you implemented.
8. Programs should be written in python with OpenCV.
9. Follow the submission instructions of assignment 1.
10. Again: you only need to implement ONE of the options.

Corner detection

1. Load and display two images containing similar content. In each image perform the following:
   - Estimate image gradients and apply the Harris corner detection algorithm.
   - Obtain a better localization of each corner.
   - Compute a feature vector for each corner point.
   - Display the corners by drawing (in color) empty rectangles over the original (grayscale) image centered at locations where corners were detected.
2. Using the feature vectors you computed match the feature points. Number corresponding points with identical numbers in the two images.
3. Interactively controlled parameters should include: the variance of the Gaussian (scale), the neighborhood size for computing the correlation matrix, the weight of the trace in the Harris corner detector, and a threshold value.

Line detection

1. Detect edge pixels so that you have a binary edge image.
2. Apply the Hough transform to detect straight lines.
3. Determine the edge pixels belonging to each detected line and refine the line parameters estimate using least squares error fitting.
4. Using color draw detected line segments and color the pixels belonging to them.
5. Interactively controlled parameters should include: an edge detection threshold parameter controlling the number of edge pixels detected, the bin size in the parameter (Hough) plane, and the peak detection threshold.
6. Add a mode for displaying the results before and after the least squares refinement overlaid in color over the original (grayscale) image or the binary edge image.
7. Add a mode for displaying the parameter plane (normalized to [0..255]).

Ellipse fitting

1. Continuously capture images from the camera. Move your hand in front of the camera.
2. Compute the difference between subsequent frames to identify pixel where change occurs.
3. Perform edge detection on the original camera and find the intersection between the edge image and pixels where changes occurred.

4. Fit ellipses to the masked edge image and display the detected ellipses on top of the captured images. Your goal is to fit ellipses to the fingers and hand.

5. Allow for interactive modification of parameters used in your program.

Active contours

1. Allow the user to specify the initial position of the input curve through mouse interaction. Implement the greedy algorithm for minimizing the energy functional. Show the evolution of the active contour until convergence.

2. Interactively controlled parameters should include: a time delay to be used between iterations, the continuity/curvature/external energy coefficients, and the fraction of points that has to reach a local minimum to determine convergence.

3. Display the results overlaid in color over the original (grayscale) image.