Computer Vision & Pattern Recognition – Spring Semester 2019

Prof. Michael Bronstein, Federico Monti, Jan Svoboda

Assignment 3

Date of submission: 16.06.2019.

If you have any questions, please send an email to the TAs. (firstname.lastname@usi.ch)
The TAs have to be able to run all the code in your submission in order to be able to grade it.

Exercise 1 - Segmentation (2 pts)

Using provided Matlab code, fill in missing body of the routine kmeans.m. Use the implemented algorithm to perform segmentation of the synthetic data using the code prepared in segmentation.m with $k = 4$ and 100 iterations.

(a) Report the results. Why do we choose $k = 4$? Experiment with different k and report your findings.

(b) **(BONUS 1pt)** Describe and implement hierarchical clustering algorithm. Apply it to segment the synthetic data from point (a). Explain your choice of parameters (if there are any) and report the results.

Take the image train.jpg. Use image color space as features and perform segmentation of the image using your routine *kmeans.m* with $k = 8$ and 100 iterations, as prepared for you in the code.

(c) Report the results. Experiment with different k and report your findings.

(d) **(BONUS 1pt)** Use more sophisticated features than just the color space to perform the clustering in point (b). Report your results.

Exercise 2 - Panorama stitching (4 pts)

Use the provided Matlab code to perform the main steps of a panorama stitching algorithm. You need to fill in two missing routines before you can run the main file, which is using them.

(a) Fill in missing body of the routine *harris.m* implementing the Harris corner detector.

(b) Fill in missing body of the routine *ransac.m* implementing the Random Sample Consensus (RANSAC) algorithm.

(c) Report your results of Harris corner detection (visually) as well as results of RANSAC algorithm (visually). If you change parameters of any routine in the file panorama.m, document it and explain why.

(d) **(BONUS 2pt)** Replace the naive keypoint descriptor (simple_descriptor.m) by implementing the Scale Invariant Feature Transform (SIFT) keypoint descriptor. Rerun the code in panorama.m and report your new findings.

(e) **(BONUS 2pt)** Use the homography estimated by RANSAC (matrix H) to transform the input images and stitch them together into a single panorama image. Report your result.

Exercise 3 - Face recognition with PCA (4 pts)

Use the provided Matlab code to perform dimensionality reduction and simple face recognition with PCA. You need to fill in missing routines *pca_fit.m*, *pca_svd.m*, *pca_eigen_decomp.m*, *pca_transform.m* and *pca_reconstruct.m*.

(a) Fill in the missing body of *pca_fit.m*, *pca_svd.m*, *pca_eigen_decomp.m*. Use *pca_fit.m* with SVD to perform PCA on the provided input data using the prepare code. Visualize the first 10 principal components. Plot the dependence of the captured variance on the number of principal components kept. How many principal components do we need to capture at least 90% of the variance?

(b) Fill in missing body of *pca_transform.m* and *pca_reconstruct.m*. Use the routines to visualize the reconstruction of the input using only the first N principal components. Experiment with different values of N and report the results. Plot the dependence of the reconstruction error on the number of principal components kept.

(c) Taking the last part of the code prepared in *pca_face_recog.m*, perform simple face recognition based on the routines you have just implemented. Experiment with number of principal components N that you keep. How does it influence the recognition performance? Why is the performance of PCA on this task so poor?

(d) **(BONUS 3pt)** Implement Linear Discriminant Analysis (LDA). Replace PCA with LDA and perform face recognition as in (c). Did the performance improve or not? And why?