

CS6550 Computer Vision Homework #3

Due Date: 12/5 23:59

Problem 1. (50%) Image Alignment with RANSAC: The problem is to find the matching between each of the three single-book images and the input image with all the books appearing in a cluttered scene. You need to implement the RANSAC-based object detection by matching a set of SIFT features with a homography transformation between two images.



- A. (25%) Use OpenCV function or publicly available program (such as the one from <http://www.vlfeat.org/>) for SIFT interest point detection to extract the SIFT feature points from the above 4 images. Establish point correspondences between the SIFT feature points detected from the single-book images and the cluttered-book image by using the distance between the SIFT feature vectors as the matching score. Show your best point correspondence results with different distance thresholds (the number of correspondences should be larger than 500) (3 images).
- B. (25%) Apply the RANSAC program in problem 1 to find the best matching homography transformation between each single-book image and the input image by using the SIFT features computed from problem 1. Show the detection result by showing the deviation vectors between the transformed feature points and the corresponding feature points on the input image. You may need to try different parameter setting in RANSAC to find the best matching result. Compare the parameter settings in SIFT feature and RANSAC and discuss the result (4 images).

Problem 2. (50%) Image segmentation:

From time to time, we shoot photos, apply image filters, and then share on social media. Yet this time, instead of using some off-the-shelf filters, you need to implement it by yourself.



Except for the given image (*2-image.jpg*), you need to use another photo you took as an input image and name it *2-masterpiece.jpg*. The required # of outputs in red brackets below is for one input image.

- A. (10 pts) Apply K-means on the image (RGB color space) and try it with three different **K** values (your **K** should be > 3) and show the results (3 images). You should use 50 random initial guesses to select the best result based on the objective function for each **K**. Please discuss the difference between the results for different **K**'s.
- B. (10 pts) Implement **K-means++** to have better initial guess (3 images). Please discuss the difference between (A) and (B).
- C. (10 pts) Implement the mean-shift algorithm to segment the same colors in the target image. Select appropriate parameters in the Uniform Kernel on the RGB color space to achieve optimal image segmentation (show the clustered result), and then show the pixel distributions in the $R \times G \times B$ feature space before and after applying mean-shift (see Unit7 p.31). (3 images)
- D. (10 pts) In addition, combine the color and spatial information into the kernel for mean shift segmentation and find the optimal parameters for the best segmentation result. (1 image)
- E. (10 pts) Show the mean-shift segmentation results with three different sets of bandwidth parameters. Discuss the segmentation results for different bandwidth parameters. (3 images)
- F. (Bonus 10pts) Compare the segmentation results by using K-means and mean-shift algorithms and their computational cost.

Reminder

- You should not use any function which can generate the result directly in each steps.
- Your code should display and output your results so that we can judge if your code works correctly.
- You should provide a **README** file about your execution instructions.
- Please compress your **code, input images, result images, report** and **README** in a zip file named **HW3_{Student-ID}.zip** and upload it to eeclass.
- If you encounter any problem, please post your problems/questions **on eeclass**.
- Please follow the file structure below:

HW3_{Student-ID}/

├ 1/

├ output/

└ // save your output images here

├ 1-book1.jpg

├ 1-book2.jpg

├ 1-book3.jpg

├ 1-image.jpg

└ HW3-1.py (or HW3-1.ipynb)

├ 2/

├ output/

├ image/

└ // save your output images for 2-image.jpg here

├ masterpiece/

└ // save your output images for 2-masterpiece.jpg here

├ 2-image.jpg

├ 2-masterpiece.jpg

└ HW3-2.py (or HW3-2.ipynb)

├ report.pdf

└ README.md