Intro to Image Understanding (CSC420)

Assignment 3

Posted: Feb 27, 2019 **Submission Deadline : March 7, 11.59pm, 2019**

Instructions for submission: Please write a document (either pdf, doc, etc) with your solutions (include pictures where needed). Include your code inside the document.

Max points: 15

1. For this exercise you will use the Scale-Invariant Feature Transform (SIFT) for matching. You will extract SIFT features from two images and use them to find feature correspondences and solve for the affine transformation between them. Please include code under each question. You are allowed to use existing code for SIFT key-point and descriptor extraction (but not for matching). Possible code to use:

   - Download code from `http://www.robots.ox.ac.uk/~vedaldi/assets/sift/binaries/` or `http://vision.ucla.edu/~vedaldi/code/sift.html` or `https://docs.opencv.org/3.1.0/da/df5/tutorial_py_sift_intro.html` There are also other packages available online, and you are free to use any of them.

   (a) **[1.5 points] Feature extraction:** Compute SIFT features for reference.png, test.png, and test2.png. You can use e.g. function SIFT() to output, for each image, a list of feature descriptors and a list of their corresponding frames. Please visualize the detected keypoints on the image. By visualize we mean: plot the image, mark the center of each keypoint, and draw either a circle or rectangle to indicate the scale of each keypoint. For clarity, please plot only the first 100 keypoints. You can help yourself with the function PLOTSIFTFRAME included in the link to the code above. (If you are using Matlab, remember to use ADDPATH() if you have unpacked the SIFT package into a subfolder.)

keypoints

   (b) **[3 points] Matching:** Given the extracted features on reference.png and test.png, describe a simple matching algorithm to find the best feature matches (correspondences) for the features in reference.png and features in image test.png. Implement the algorithm in your favorite programming language. Visualize the top 3 correspondences. Please do the same using the test image test2.png. Show each image and visualize each correspondence by indicating the fea-

ture's position and scale in the appropriate image. Use a separate color for each correspondence.

(c) [**2 points**] **Affine transformation:** Use the top 3 correspondences from part (b) to solve for the affine transformation between the features in the two images.

(d) [**1 point**] Visualize the affine transformation. Do this visualization by taking the four corners of the reference image, transforming them via the computed affine transformation to the points in the second image, and plotting those transformed points. Please also plot the edges between the points to indicate the parallelogram. If you are unsure what the instruction is, please look at Figure 12 of [Lowe, 2004].

2. [**2.5 points**] Take an item for which you know the width and height (in cm), for example a piece of paper or a dollar bill. Place the item next to or in front of the door. Take a picture of the door such that all four corners of the door are visible on the photo. Take this picture in an oblique view, ie, the door is not a perfect rectangle but rather a quadrilateral in the photo. Estimate the width and height of the door (in cm) from the picture.

3. [**5 points**] You are given a few photos of landscape. The goal is to take two photos, LANDSCAPE_1 and LANDSCAPE_2 and stitch them into one photo. You can do this by extracting SIFT features from both photos, match them, and estimate a homography of one photo to the other. Use RANSAC to find the best homography. Once you compute the homography, "stitch" the two photos together, forming a small panorama. We will give half points if you compute affine transformation instead of a homography.