Image Registration Project
ECE 278: Fall 2019

B.S. Manjunath
manj@ece.ucsb.edu

October 9, 2019

1.1 Introduction

In this project you will implement an end-to-end image registration system. Your system must be capable of aligning image pairs to a common coordinate system. The project is divided into different steps (roughly one for each module of the system).

Peer discussion is encouraged and you can use online resources, including code samples, as needed. However, you will need to clearly acknowledge all the sources and must submit the source code.

The project report is free-form, this being a graduate class. Make sure that you emphasize all of the critical pieces and adequately address all the issues below.

1.2 What to Submit

Before due date, you will upload ZIP folder that should contain the following items:

- Your matlab code in a folder called `YourLastNames_registration`
- A report file called `report.pdf` containing a *brief* description of all the functions that you wrote and the details that are requested for the different parts (see next section).
- The images containing the final result of you registration software (i.e., the input images aligned to the same coordinate system) in a directory called `results`

For full credit, these must be uploaded before the due date.

1.3 Project Parts

Part 1: Point Feature Detection

Given the image pair that is to be registered, identify a set of tie points in each image. This will be combination of significant feature detection and localization, and at the minimum should have a SIFT-like implementation.

You can use any online codebase to compute the SIFT features or your own implementation. Please keep in mind that it will be difficult to debug other people code, so important that you choose the right ones.

Part 2: Establishing Correspondences

Implement a feature descriptor routine that associates to each point a descriptor (e.g., SIFT). Once this is done, compute the putative correspondences using a suitable distance function. It is likely that you will obtain a large number of mis-correspondences: you will take care of this problem later on. In the report explain clearly the details of your feature descriptors and the procedure that you adopted for creating the putative correspondences. *This probably is the toughest part in automating the registration. Finding good corresponding points is not easy.*

Part 3: Estimating the Homography

Implement the standard DLT and the normalized DLT algorithms to estimate the parameters of the homography that relates the image pair. In the report write explicitly the homography that you estimated for each image pair. To facilitate the grading write the matrix H after normalizing its coefficients so that $H_{3,3} = 1$.

Establish the correspondence between image pairs and write down the H matrix for both the un-normalized and normalized DLT.

Part 4: RANSAC

Set up a RANSAC framework to estimate the homography in presence of outliers. In the report include the same information (the homography) required for the mandatory part. Write down the H matrix after the RANSAC pruning step.

Part 5: Image Warping

Warp the images to the same coordinate system and then superimpose them so that the quality of the registration can be clearly appreciated *visually*. These images must be included in the `results` directory of the archive you will submit.

You may find inspiration checking the matlab function `imtransform`.

You should submit three cases for each pair of images. (a) Standard (unnormalized) DLT, (b) Normalized DLT, (c) Normalized DLT + RANSAC.

Useful online resources

1. VLFeat Library: see `http://www.vlfeat.org/`
2. OpenCV: `http://opencv.org/`
3. Peter Kovesi Matlab Image Processing functions: `http://www.peterkovesi.com/matlabfns/index.html`
4. Marco Zuliani's RANSAC toolbox `https://github.com/RANSAC/RANSAC-Toolbox`