ECE 278A

Date: 31 October 2017
To: Dr. Manjunath
From: Austin McEver and Christian Lee
Subject: Project 1 Report

This report describes each function in the source code of our project one. It also includes output images computed by our project, and lists webpages we used to help with this project.

Overview

Within this project there is an ImageProcessor class, which handles all the heavy lifting. Using it are two scripts, ImageWarper.py and ImageMosaic.py. Both scripts take two command line arguments: the path to two images. ImageWarper then warps the first image into the space of the second image and qualitatively should make objects in the first image like they do in the second image. ImageMosaic attempts to combine the images into one larger image such that objects shared between the two images are overlapping.

ImageWarper

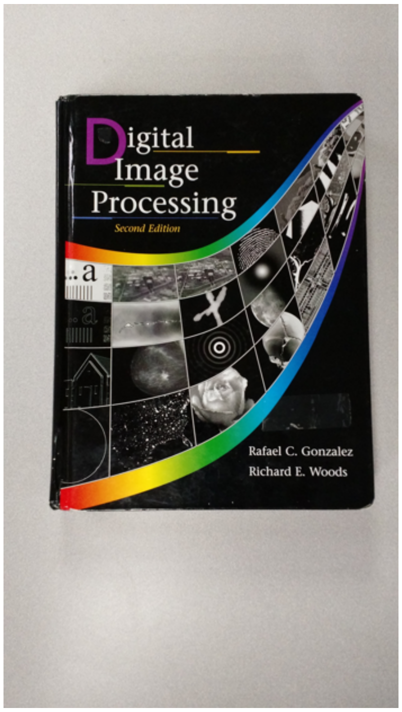
ImageWarper simply creates an instance of ImageProcessor and calls its Warp method. Warp uses OpenCV's SIFT method to find keypoints, and then feeds those keypoints into CV's brute force matcher. It then calls ImageProcessor's RANSAC method, which is a method for CV's RANSAC method. Then, depending on the options passed to Warp, ImageProcessor runs DLT, normalized DLT (nDLT), and/or RANSAC to warp the input image into the other input image. We implemented DLT ourselves using an online note resource (https://me363.byu.edu/sites/me363.byu.edu/files/userfiles/5/DLTNotes.pdf).

ImageMosaic

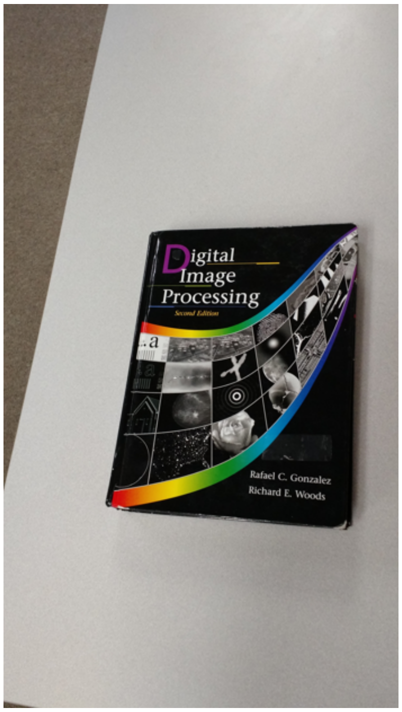
ImageMosaic works similarly to ImageWarper in that it creates an instance of ImageProcessor and calls its Mosaic method. The Mosaic method begins similarly to Warp: it runs SIFT, brute force matcher, then RANSAC. Finally, it uses openCV's warpPerspective method to warp one image into the space of the other. Then, it maps one image on to the other using Python list replacement. Finally, it shows the result.

Results

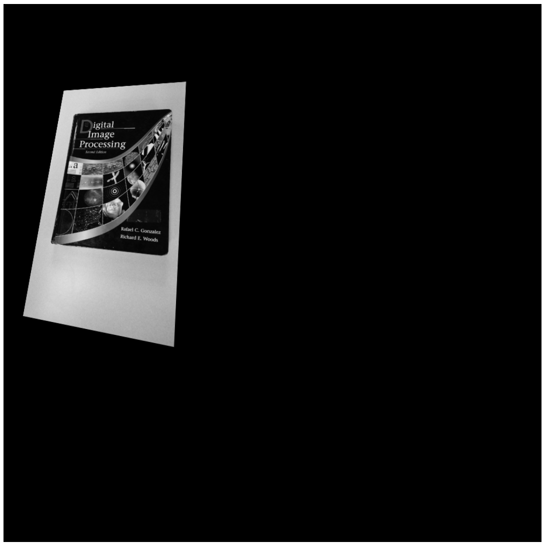
The following images show the output of the ImageWarper and ImageMosaic each with one pair of input images. ***show keypoints as well***



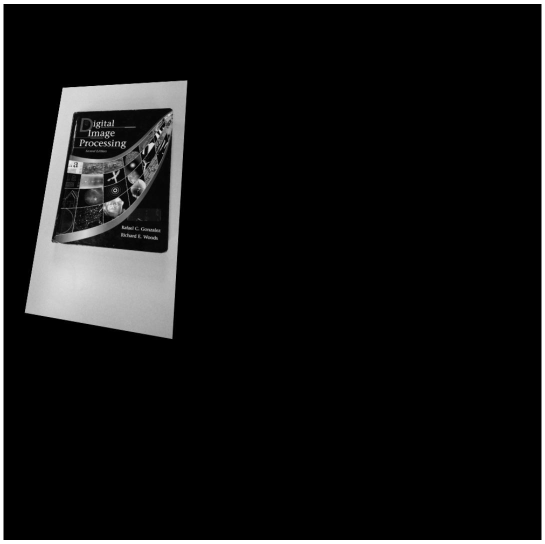
Warper input image 1



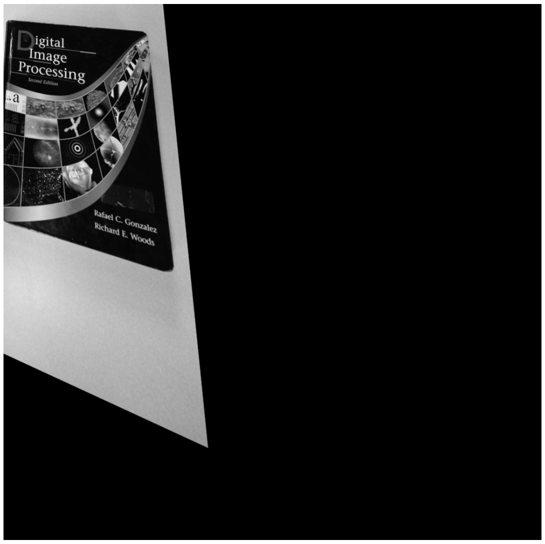
Warper input image 2



Warper DLT output



Warper RANSAC output



Warper normalized DLT output

UIC

UIC

Conclusion

Right now, normalized DLT isn't working that well, and our mosaicking only works in a select few cases where no rotation is required. Should be fixed in the next upload. Cheers.

Sources

- https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/
- https://docs.opencv.org/3.3.0/db/d27/tutorial_py_table_of_contents_feature2d.html
- Fischler (RANSAC) - https://gauchospace.ucsb.edu/courses/pluginfile.php/1416851/mod_label/intro/PaperpFischler1981.pdf
- Lowe (SIFT) - https://gauchospace.ucsb.edu/courses/pluginfile.php/1416851/mod_label/intro/Paper_SIFT_IJCV2004.pdf
- https://www.youtube.com/watch?v=oT9c_LlFBqs&t=2173s
- https://me363.byu.edu/sites/me363.byu.edu/files/userfiles/5/DLTNotes.pdf

hello type here