Andy D. Martinez
HW3 report

How To Run The Code:
Run the hw3.m file.

Input:
The file hw3.m keeps a variable called directories in which it stores all the directories that contain frames to be used for the purpose of building a panoramic image out of them. There is no command line input needed. In order to remove or add a new directory with frames to create a panoramic image, add the name of the directory on the directories cell array.

Output:
For each directory of images named d, the program produces an image called d_pano.jpg which is the panoramic image created using the images inside the directory named d.

My Files:
hw3.m : the main file to run. It iterates over each directory name on the cell array directories and saves the panoramic image created out of directory inside directories.

hw3_helper : It is the file that calls helper functions to load images, create homographies, transformations, apply transformations and builds the panoramic image.

load_images.m : loads all images on a directory with name provided as input.

homography.m : Given 2 images, it finds SIFT features, and matches between them.

create_panorama.m : Takes the current panoramic image, the next image to add and its transformation and adds it to the panoramic image creating a new panoramic image.

Imported External Files:
ransacfithomography.m, ransac.m, normalise2dpts.m, iscolinear.m, homography2d.m, hnormalise.m

Out of those files I am only using ransacfithomography.m, however, it depends on all the other files.

Required Libraries:
To calculate SIFT features, I am using the open source library vlfeat-0.9.20.

Method:

Taking as starting point the first frame, I started by finding the homography of each pair of consecutive images. In order to find that homography, SIFT descriptors were found for each image and then matches of those descriptors were found for each pair of consecutive frames. Then those matches were used as input to RANSAC to find the homography between the two frames. RANSAC used a threshold of 0.01 which was on the recommended range and after exhaustive testing seems to provide good results.

A homography was then constructed for each frame by multiplying the transposed homography obtained for that pair, with the ones of the following pairs, carrying information all the way from the last frame. This way, the infinite homography of each frame associated with the last frame was constructed.

The dimensions of the panoramic image were found by finding the difference of the max and min x and y limits.

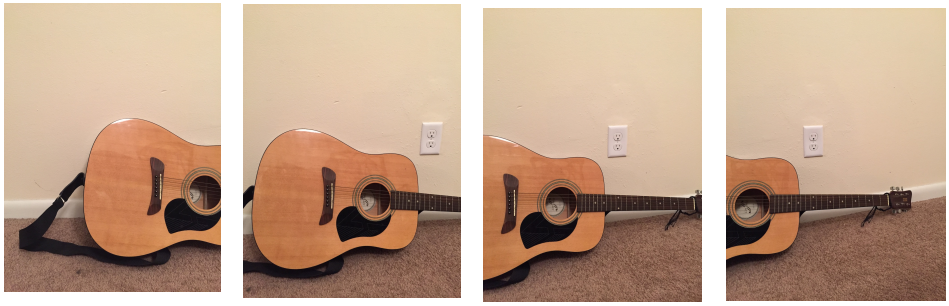
Each frame was then warped using the previously-created transformation corresponding to the infinite homography of that frame associated with the last frame of the sequence. The function imwarp() from Matlab was used. The 'OutputView' option was used for imwarp, which warps the original frame into its specific location in a world coordinate system. The world coordinate system was set by using a previously-created 2D spatial referencing object that contained the dimensions of the output panoramic image.

Then the panoramic image was found by blending the current panoramic image with the warped version of each frame. This blending was done by using the alpha blender (vision.AlphaBlender()) provided with Matlab's Computer Vision toolkit.

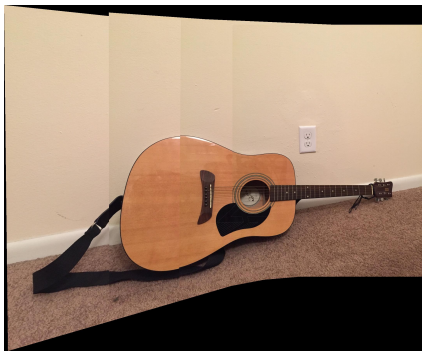
For this, I started with the panoramic image initialized with all zeros according to the dimensions found before, and used the alpha blender to blend it with the warped version of the first image.

Then, similar steps were created to blend the current panoramic image with the next warped image.

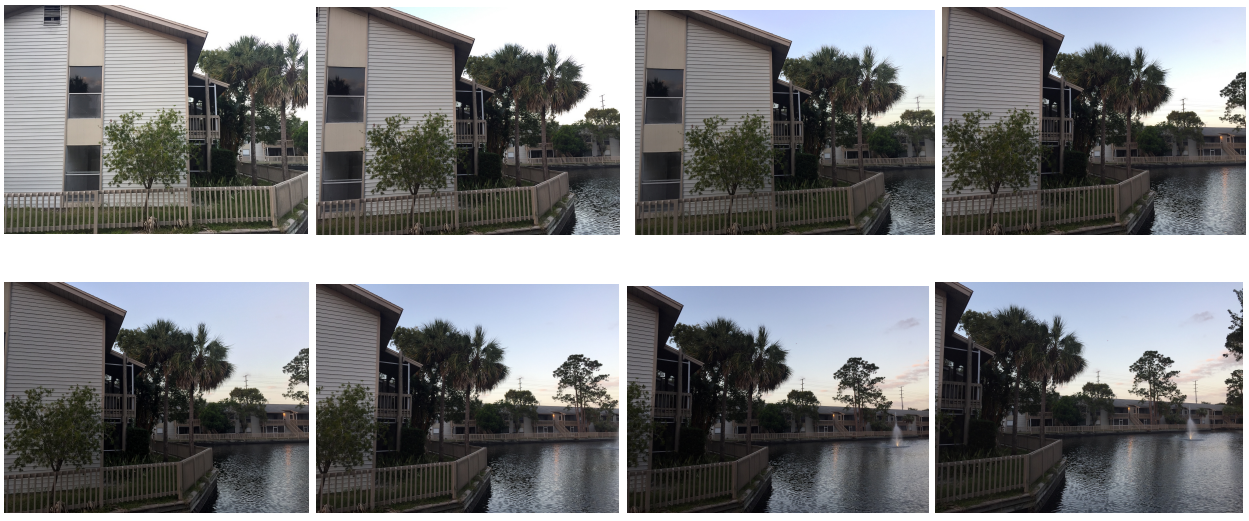
Sample Input 1:



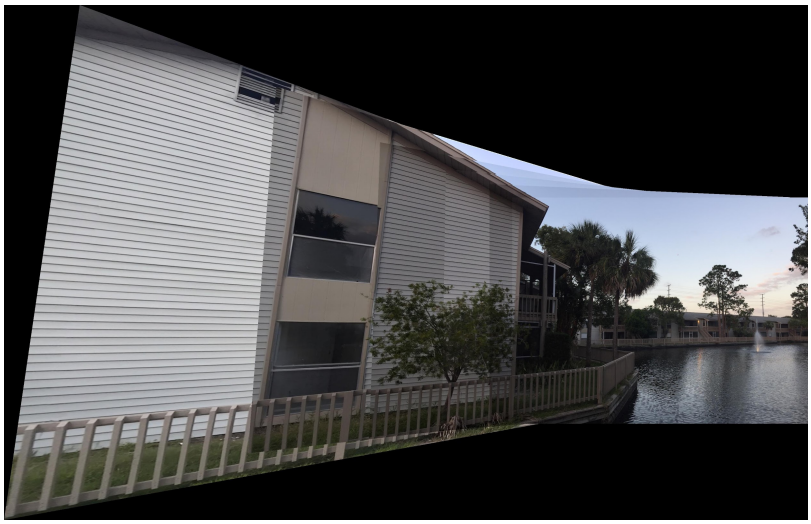
Sample Output 1:



Sample Input 2:



Sample Output 2:



The geometry of the Panoramic Image Stitching looks great. It is the illumination difference which can be appreciated on the original images which makes the output image look a little bit different since one part of the image had a lot of illumination and the other part was darker.

Output of provided 21 images on mov2 directory: