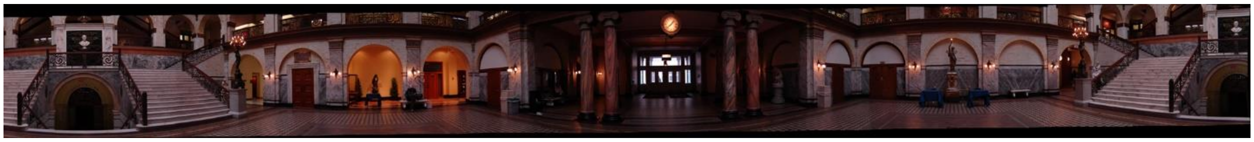

Figure 1. Example output recreated by my script.

Above is the final output panorama of the 14 images provided in the example input. There were no parameters to change, so the output should always be the same. The exact displacement numbers between the images seem to change by operating system or architecture, as they are not the same as the example output numbers provided. However, the slight shifts in the result apparently affects the panorama nominally (I don't see any visual differences between my output and the example output). I see some stitching areas are a little off, which is especially noticeable in the floor, but this is probably due to drift, and is apparent in the example output image as well. I tried to fix the drift by computing the homography of the image, but I couldn't figure out what points to use to warp the image properly at first. Figure 2 below is an example of an attempt.

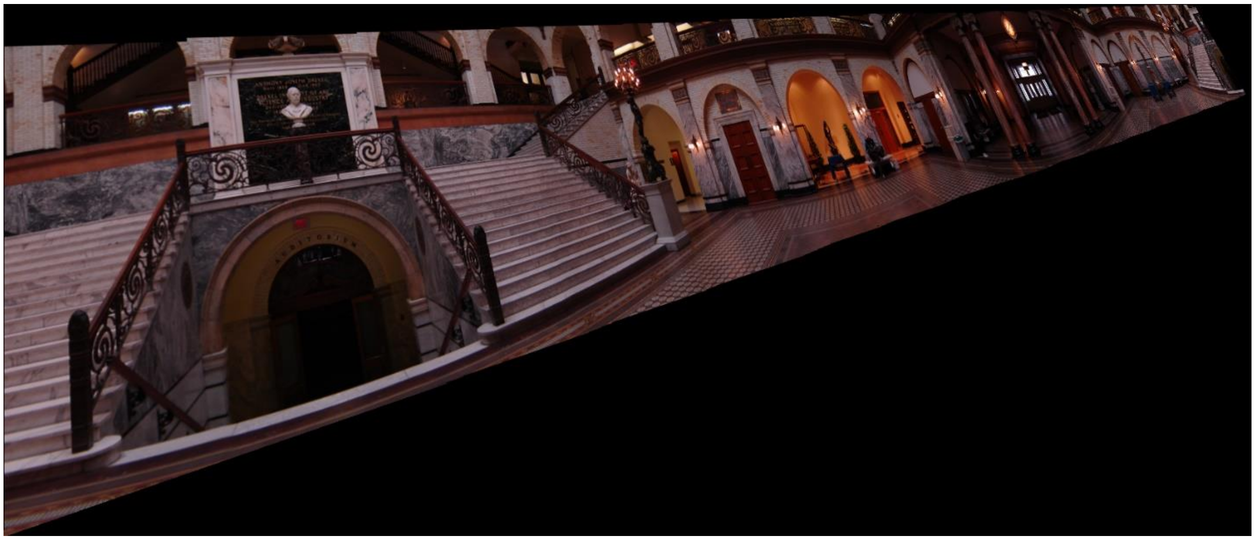

Figure 2. I tried doing the warp to fix the drift. I failed.

After some adjustments with the points I was using, I was able to compute the correct homography that fixed the drift and then cropped the image.

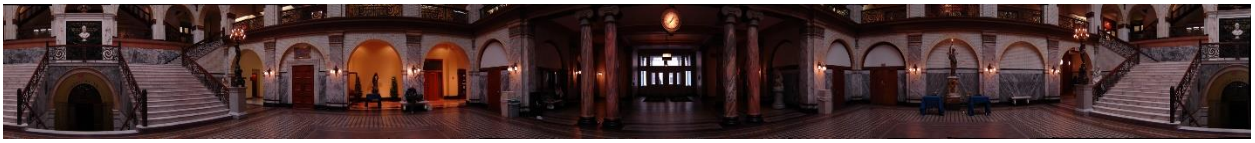

Figure 3. Correct homography applied to panorama to account for drift.

From there I needed to calculate the distance from the left image to the first to loop the image, which was easy with the displacement I already calculated. The formula I used:

(panorama_width - (right image width – abs(displacement of leftmost and rightmost image))))

gives the amount of pixels from the right of the image to delete.

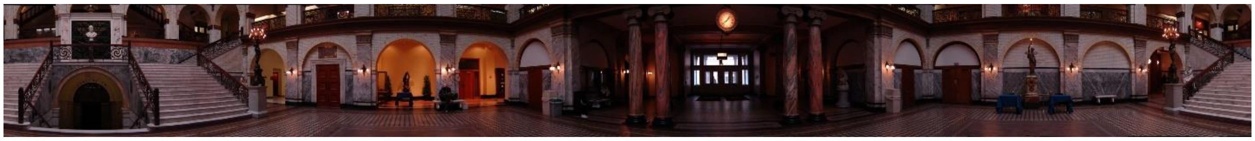

Figure 4. Final panorama, warped, cropped, and looped.

The issues I had with this project all resulted from floating point math. I spent some time trying to figure out why none of my Lucas-Kanade tests were passing, only to realize they passed on Tux (I was compiling on Windows). Now I had to transfer my files each time I wanted to run a test. Now running on Tux, I was still unable to pass the pyramid Lucas-Kanade, even though I was very, very close, but I assumed I was doing something wrong that was affecting my answer, or it was a coincidence that it was so close. This was mainly due to me not being confident in the algorithm: should I be scaling by two? should I be accumulating displacements? (this was unclear). However, I found a paper describing the pyramid LK algorithm in detail (http://robots.stanford.edu/cs223b04/algo_tracking.pdf) which confirmed that these slightly differences were apparently okay.