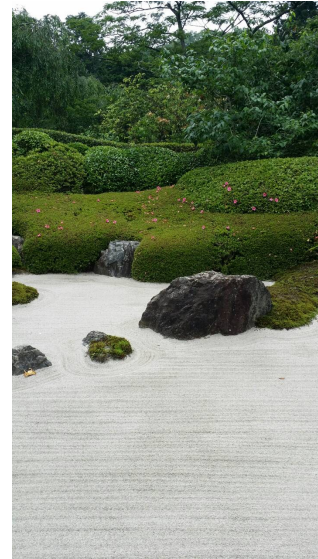


Computational Photography

Jacob Fund

Assignment 8

Input Images



Zen Garden from Kamakura, Japan, and a little golden crab is peacefully meditating.

blendImagePair

Originally I was trying to implement this by computing Laplacian pyramids. This proved to be difficult and I ran into pixels that were so off that it seemed like there was some bug in my code. So then I just implemented the simple averaging of the pixels that intersected the two pictures.

What effect did increasing the number of matched features have on your output?

When I increased the amount for the features, this created more votes for the algorithm to determine the correct homography. With too many matches that exceeded over 70, I got very warped results. This makes me think that there are many outliers within my images.

What type of panorama did you take?

This was a planar panorama. I tried my best to keep my hand straight so the images came out without any rotation.

Results



Happy with the results?

No, I spent a lot of time messing with the number of matches and the RANSAC threshold. But I found that 50 matches and keeping the threshold at 5.0 gave the best results. If you look at the rock in the middle you can tell that it did not align correctly. A seamless blend would have been nice. I was close using the pyramid strategy, if I had more time I would look into implementing this.

Note: We'd like you to explain the reasoning behind multiplying the x_{\min} and y_{\min} by negative 1 in your writeup.

I believe by making the x_{\min} and y_{\min} negative the homography will correct itself for a negative minimum for the new image to be warped onto.