Georgia Tech's Computational Photography Portfolio

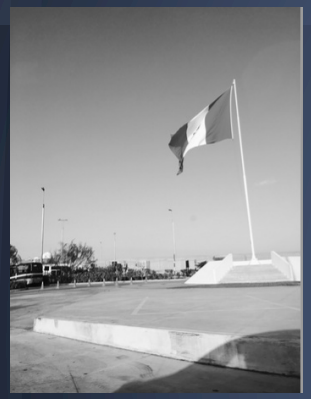
Ram Subramanian
ram.s@gatech.edu

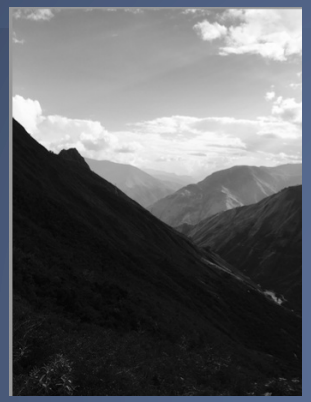
https://cs6475.wordpress.com/fall-2016/

Assignment #1: A Photograph is a Photograph

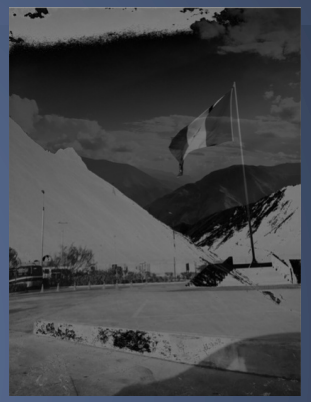
Salkantay peak, Peru
Using an iPhone 6S

Assignment #2: Image I/O

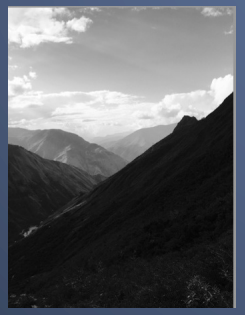




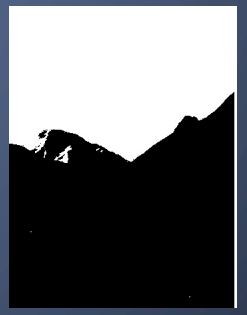
averageTwoImages



flipHorizontal



convertToBlackAndWhite

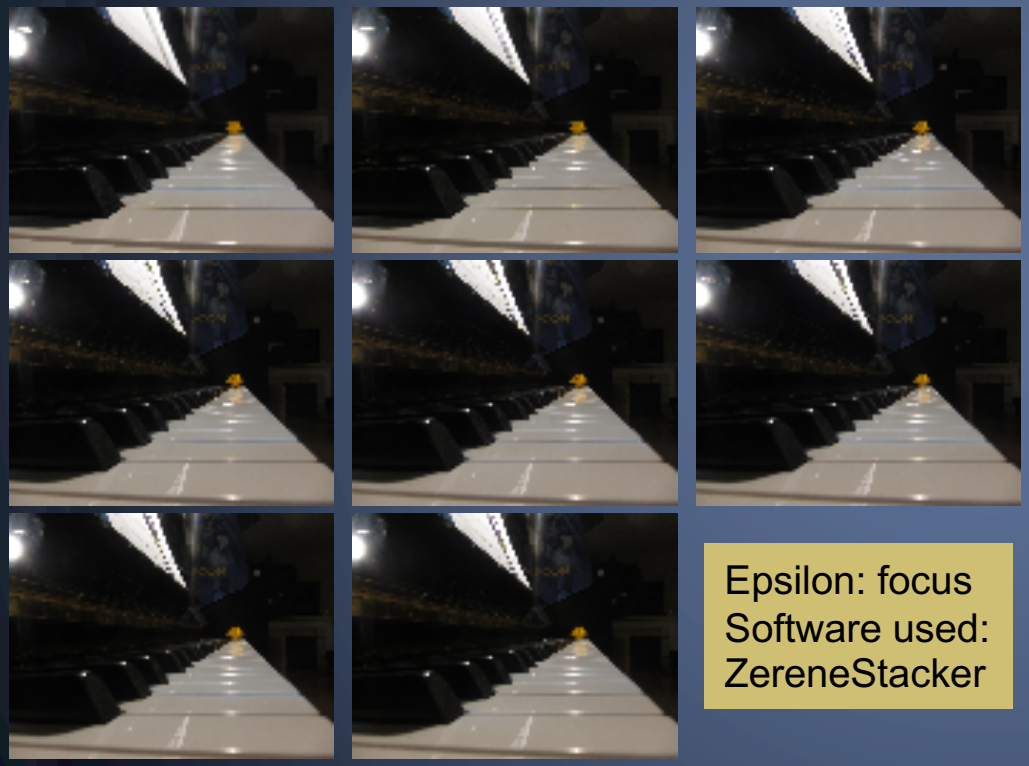


Assignment #3: Epsilon Photography

The objective of this assignment was to take multiple pictures of a scene varying just a single variable each time, and to produce a novel artefact from those images.

I chose the focus as my epsilon for this assignment and the scene constituted of the length of the keys of my piano and a small yellow puzzle at the end.

Assignment #3: Epsilon Photography



Epsilon: focus
Software used:
ZereneStacker

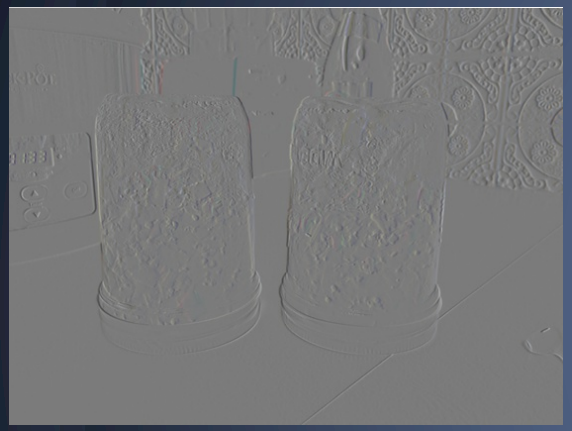
Assignment #4: Gradients and Edges

The objective of this assignment was to write an algorithm to produce an edge image of a given input image.

I chose two bottles of hot sauce I made against my kitchen counter-top as my scene, as it provided for ample edges.

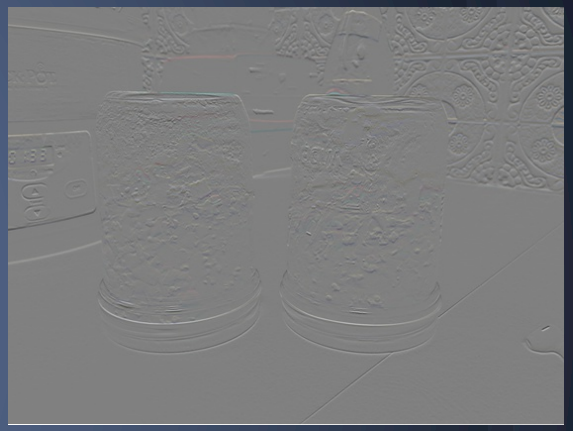
I was quite satisfied with the results as my image was quite close to the one produced by the famed Canny edge detector (minus edge-thinning).

Assignment #4: Gradients and Edges



gradientX

gradientY



my edge detector

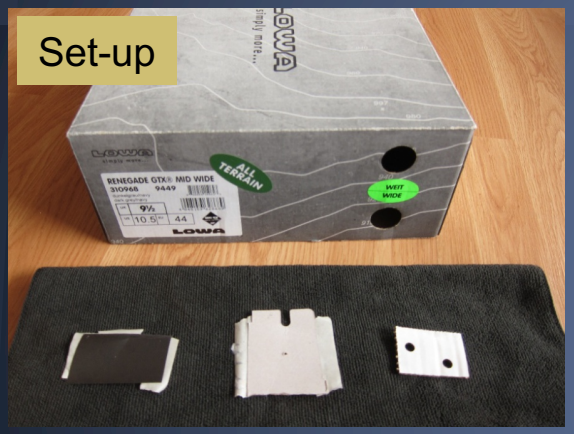
cv2.Canny

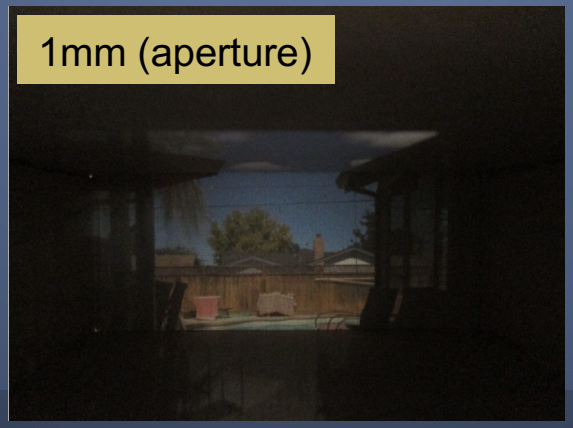
Assignment #5: Camera Obscura

The objective of this assignment was to build either a room or a box obscura and learn about the basic principles of a camera though experiments on the aperture.

After much difficulty in getting my room-obscura to work, I build a box-obscura out of a shoe box and holes of different sizes punched onto thin pieces of cardboard.

Assignment #5: Camera Obscura





Assignment #6: Blending

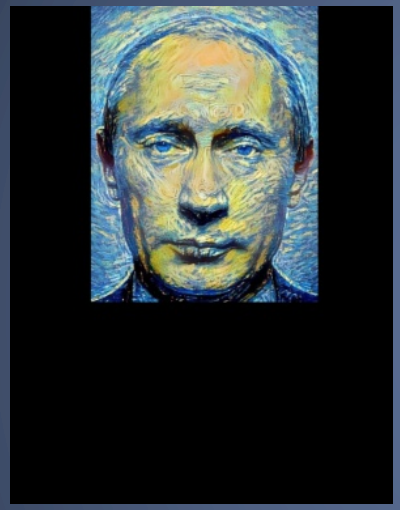
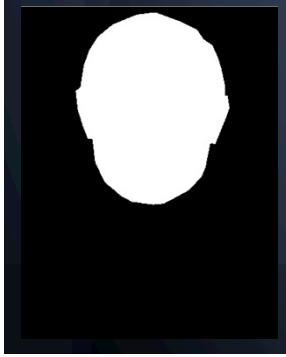


Image 2

Pyramid depth = 1

Pyramid depth = 5

Pyramid depth = 8



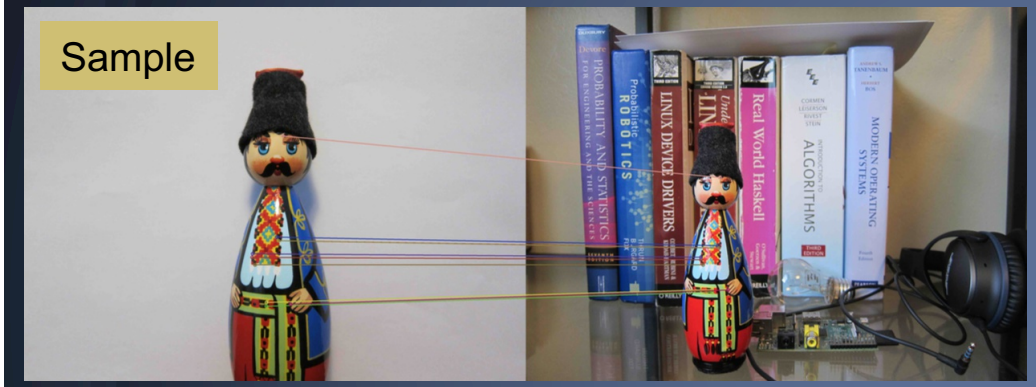
Blending mask

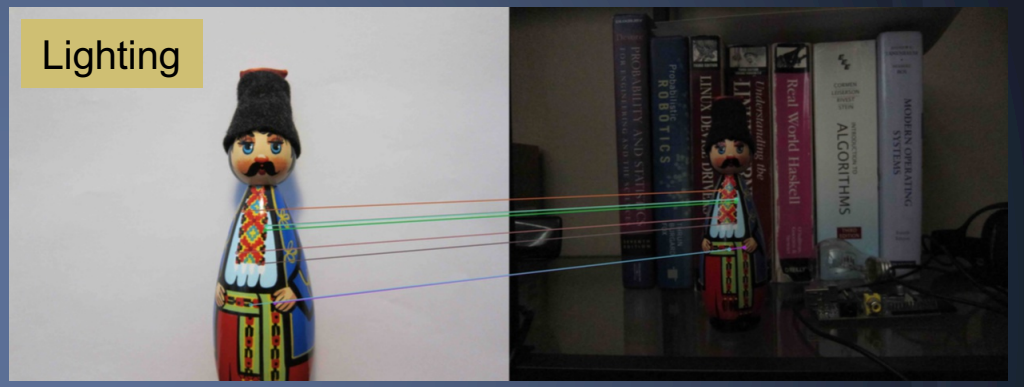
Assignment #7: Feature Detection

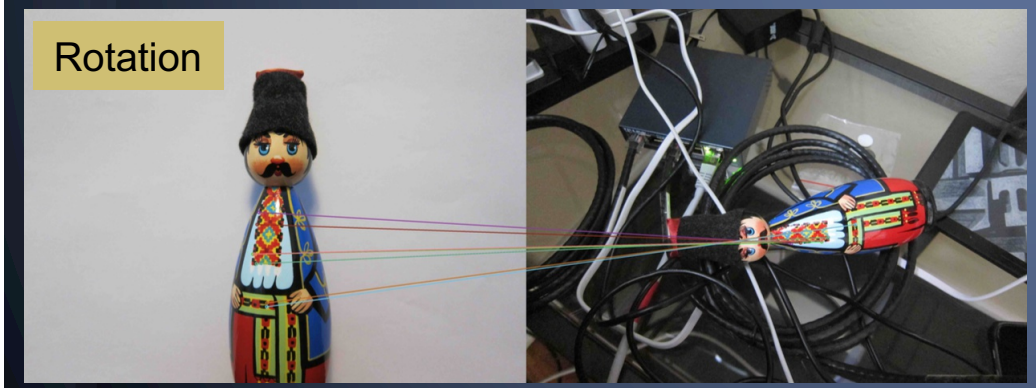
The objective of this assignment was to use the ORB feature detector and match features between images, used in this case to find an object, but could be used for many other purposes – such as generating a panorama.

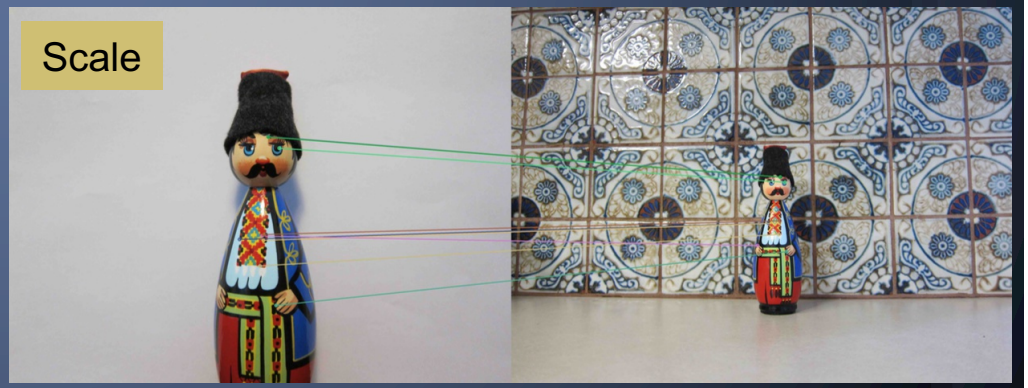
I used a doll I keep on my shelf as the subject for this assignment due to the amount of features it could provide – and it did very well on the tests (10 on 10 everytime!)

Assignment #7: Feature Detection

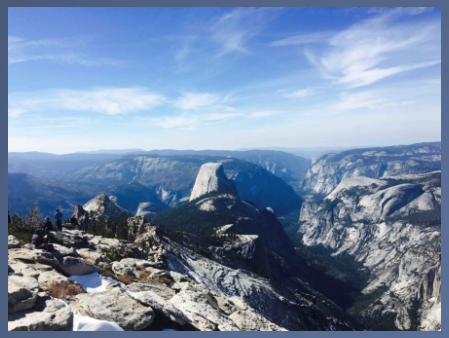


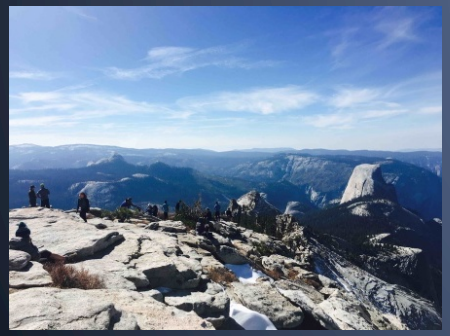






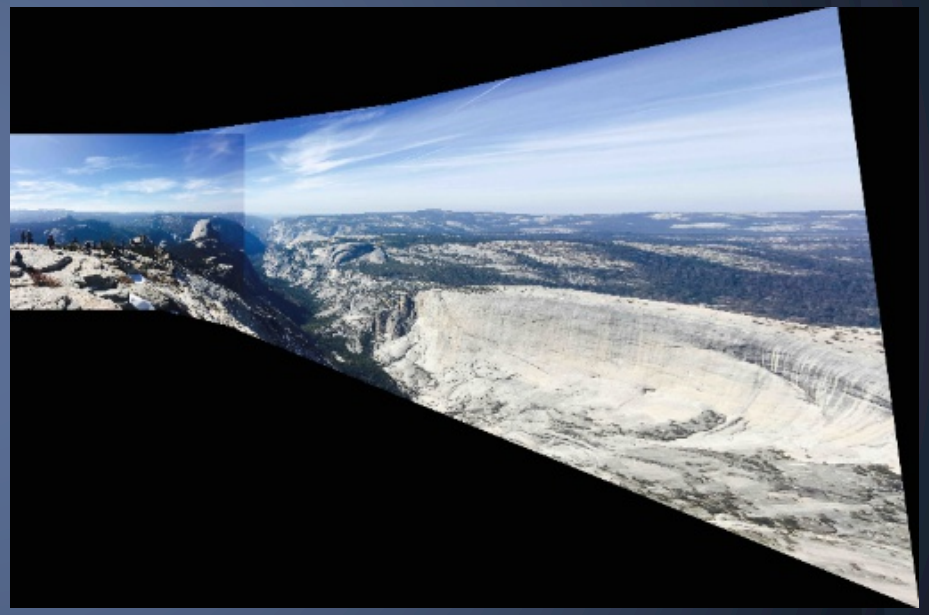
Assignment #8: Panoramas





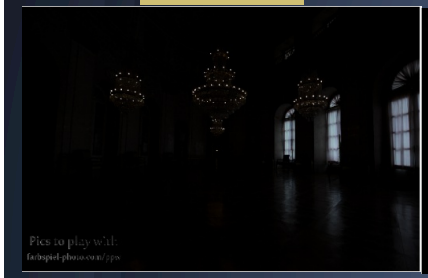
A view of half-dome from Cloud's Rest

I would liked to see a bit less of warping, but I attribute this to me not exactly pivoting about a single axis.



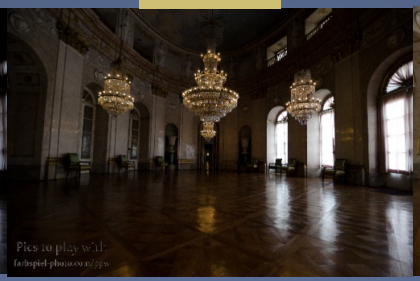
Assignment #9: HDR

1/2500



1/640

1/160



1/40

1/10

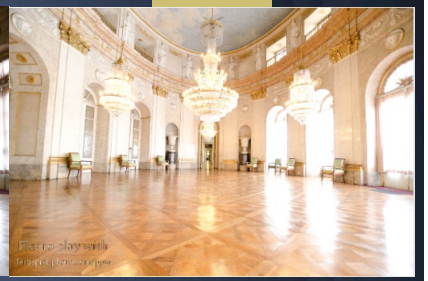
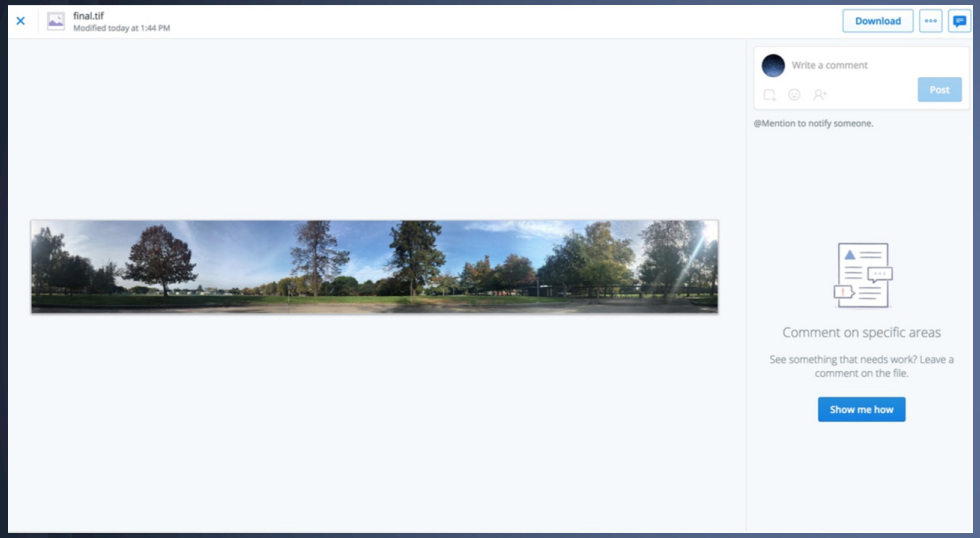
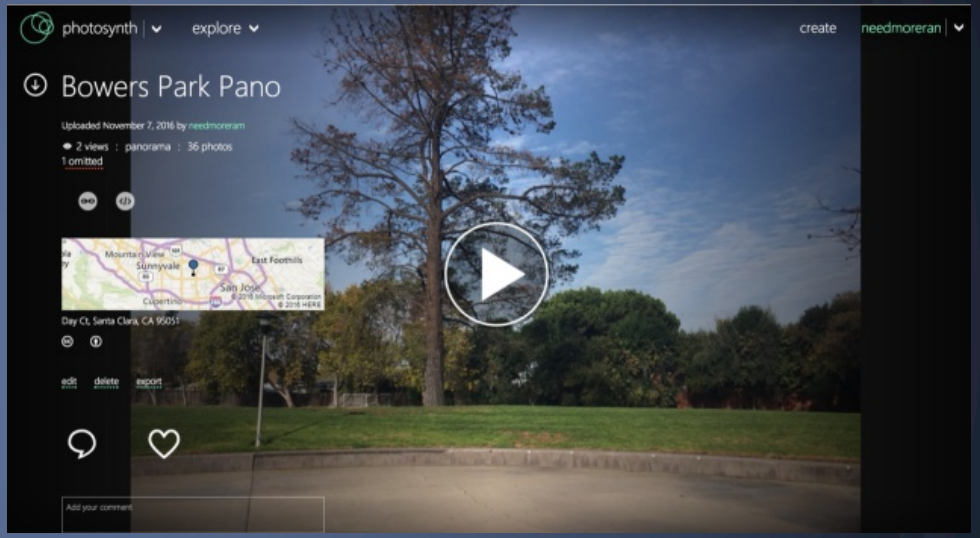


Photo credits:
Farbspiel photgraphy

Final HDR image

Assignment #10: Pictures of Space (1)

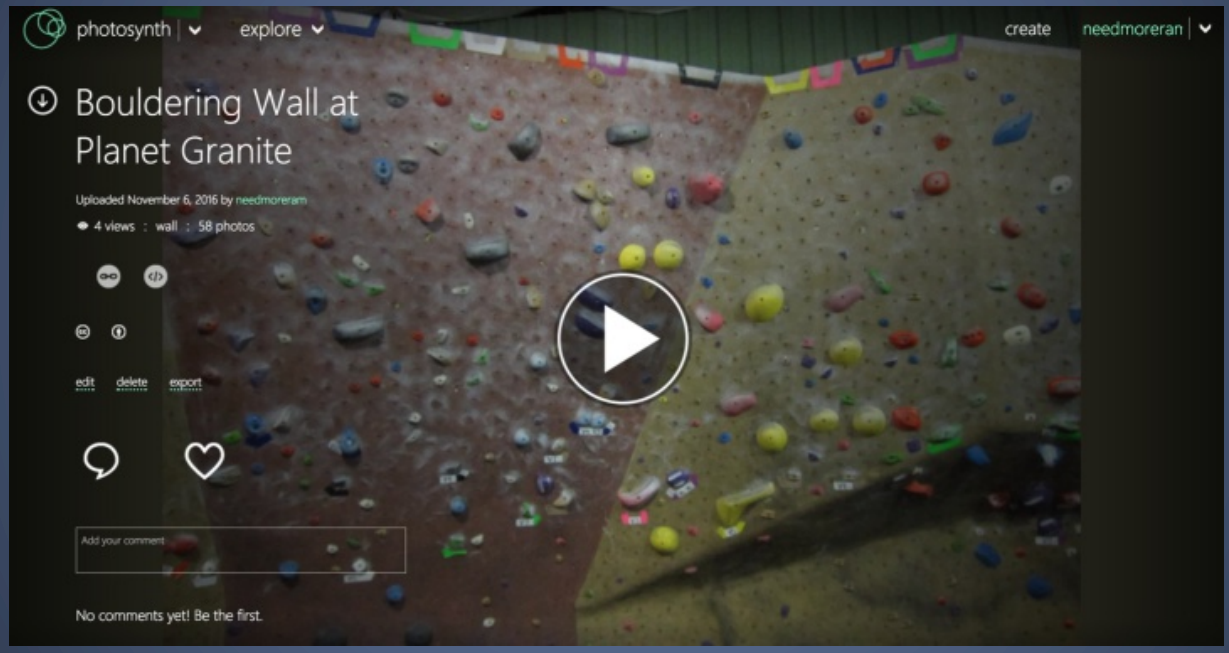




Panorama of Bowers park
Created using Hugin
https://dl.dropboxusercontent.com/u/14887563/final.tif

Another panorama of Bowers park
Created using Microsoft Photosynth
https://photosynth.net/preview/view/6a80bcca-f4cc-4fde-8df8-71121cd928ed

Assignment #10: Pictures of Space (2)



A bouldering wall at Planet Granite, Sunnyvale, CA
Wall synth, created using Microsoft Photosynth
https://photosynth.net/preview/view/a1611419-f740-4ab1-ba82-520f72786ca4

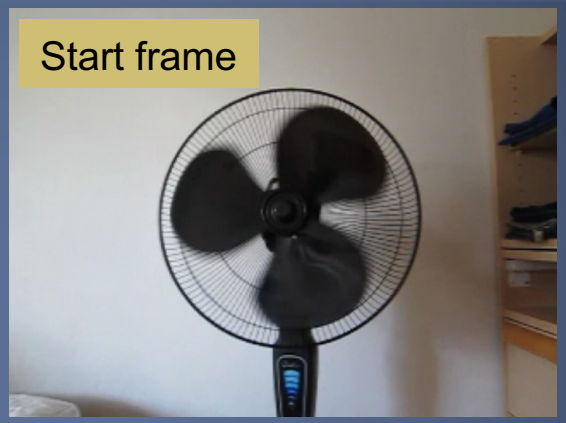
Assignment #11: Video Textures

Diff1

Diff2

Diff3

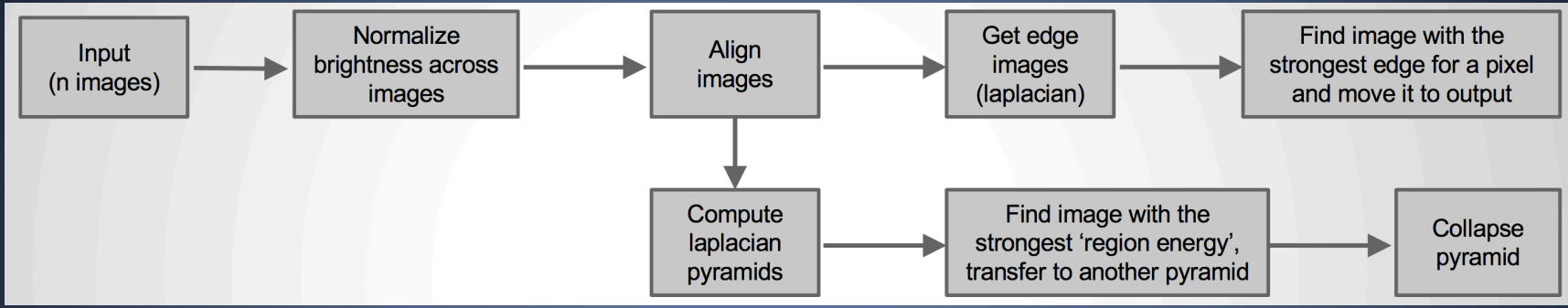
Start frame



End frame

Final texture:
http://i.imgur.com/VCBSg74.gifv

Final Project



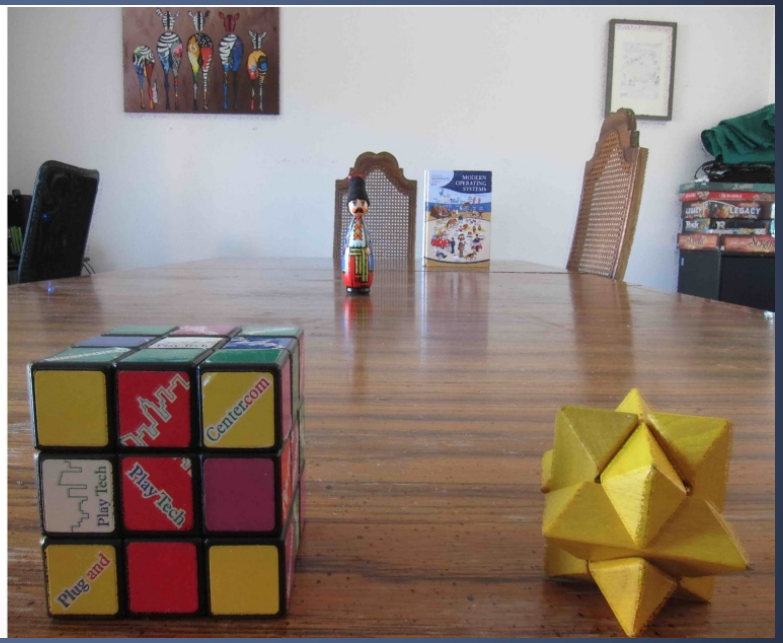
For my final project, I wanted to explore different ways to stack photos of a scene taken with different focal lengths into a single artefact. Ideally, this final artefact would have all areas/objects within the image in-focus (inspired by ZereneStacker, a program I used in completing assignment 3).

For this I implemented two different methods: one that uses an edge image and gradients to determine the strongest edges (pixel-by-pixel) and the other using laplacian pyramids and looking at 'regions' of strong edges.

Unfortunately, I lost all of the originals so I had to get the images from my report.

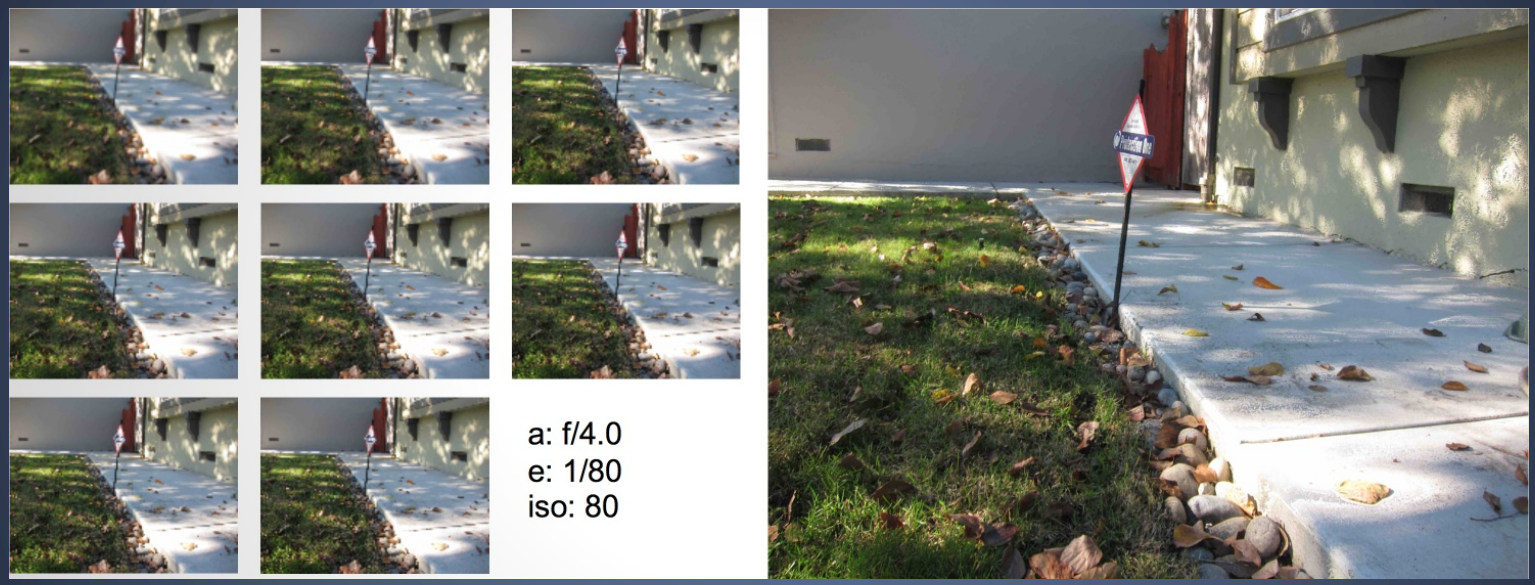
Final Project (2)

a: f/4.0
e:1/30
iso: 800



For my first test, I placed different objects at different distances from the camera.
As you can see the final result of the pixel-by-pixel method is quite good, except at the edges of the yellow object in the foreground.

Final Project (3)



For my second test, I took a few pictures of the yard sign in front of my house.
Additionally, I wanted to see how my algorithm would function when things are moving in the background (like the shadow of the tree). The algorithm again performs quite well (you can even read the yard sign!), but the area where the shadow moved between pictures is a big jumpy.

Final Project (4)

a: f/3.0
e: 1/50
iso: 400

Output

For my final test, I wanted to test my algorithm against pictures taken by another camera (one that doesn't exhibit zooming effects when changing focal places, etc).
And I'd say the result is quite flawless :)