Georgia Tech's Computational Photography Class Portfolio for Spring 2016

J. Michael McGarrah
https://blog.mcgarrah.org / mcgarrah@gatech.edu

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

Assignment #1: A Photograph is a Photograph

Demonstrate a Digital Photograph that you have taken.

"Sunrise at Family Farm"
Crown Point, IN, USA
October 2013
Using Samsung Galaxy Note II cell phone

Assignment #2: Image I/O

Process a digital image at the pixel level.

Merging of two gray scale images

"Daughter in China"
Various locations, China
December 2006
Using Nikon E3100 camera

Assignment #3: Epsilon Photography

Demonstrate a photo produced from several slightly varied source photos.

Five photographs of a GaTech coffee mug moving across a space with shadows and reflections.
In the merged final image below, all artifacts of the coffee mug are missing.

Raspberry Pi 2
5MP Webcam

```
   for x in range(size_x):
       for y in range(size_y):
           for ch in range(size_ch):
               # get a color value from pixel channel
               bgra[ch] = img.item((x, y, ch))
               # append the value to the values table
               values[x][y][ch].append(bgra[ch])
print('Begin merging image files')
# create an empty image file of same size as originals
final_image = np.zeros((size_x, size_y, size_ch), np.uint8)

# Here is the merge magic with using the median values
for x in range(size_x):
   for y in range(size_y):
       for ch in range(size_ch):
           # grab the pixel color values for each pixel color
           ch_values = values[x][y][ch]
           # find the median value of the pixels
           ch_median = np.median(ch_values)
           # write median pixel values to new image
           final_image.itemset((x, y, ch), ch_median)
```

Python Code

The average value for each pixel in the five (5) images are used to create the most common pixel from each.

Assignment #4: Gradients and Edges

Produce an edge detection process from base concepts.

Original Image

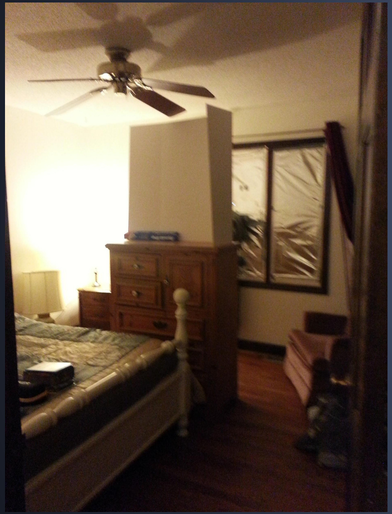
Custom Edge Detection

To the left is an implementation of edge detection for the class project with changes in both vertical and horizontal.

Edge detection is the name of a method which identifies the points in a digital image where the brightness changes sharply between pixels. Those points of sharp change are typically organized into a set of curved line segments called edges. Edge detection is a fundamental tool in image processing and computer vision, particularly in the areas of feature detection and feature extraction.

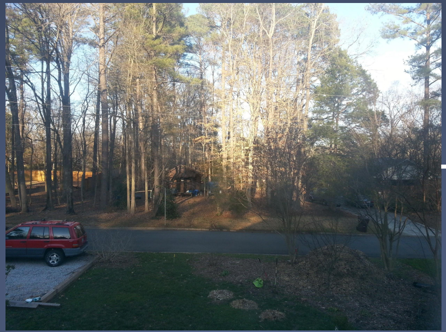
Assignment #5: Camera Obscura

Use the simple optics and produce an image. Processing image includes Raspberry Pi 2.



The Setup

Very Dark Room with tiny hole to project an image on a surface



INPUT - The Scene

OUTPUT - The Image

Raspberry Pi 2
5MP Webcam

+

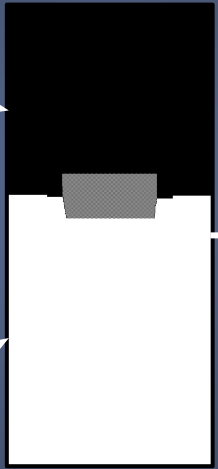
Image Magick Tools

Assignment #6: Blending

Use pixel level processing to merge two images smoothly.

Black

White



Mask

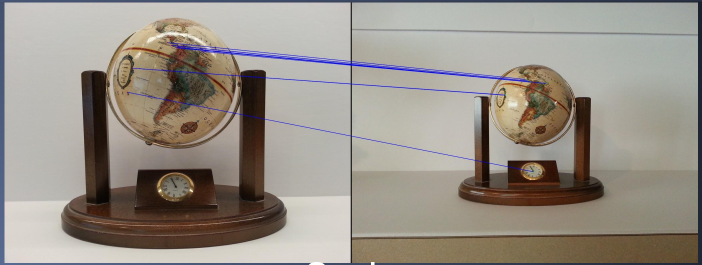
Blended

Blending done with Python image processing code.

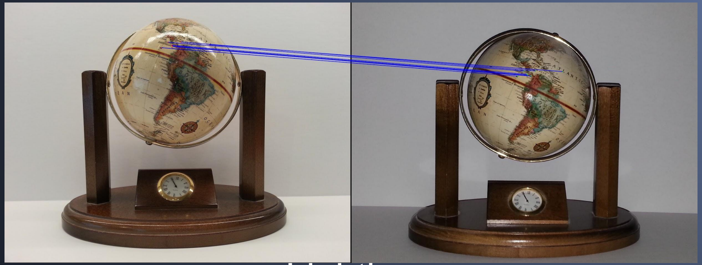
Assignment #7: Feature Detection

Detect features in common between photos and map similarities.

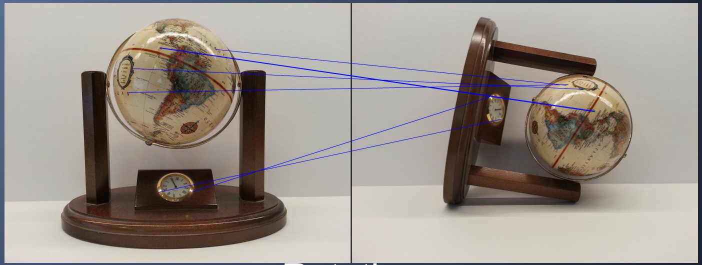
Sample



Scale



Lighting



Rotation

Feature detection is a method for taking a template image and find it inside another image. Finding and matching key points of similarity between images while changes like rotation, scaling, exposure and mixing with other objects is the goal.

Assignment #8: Panoramas

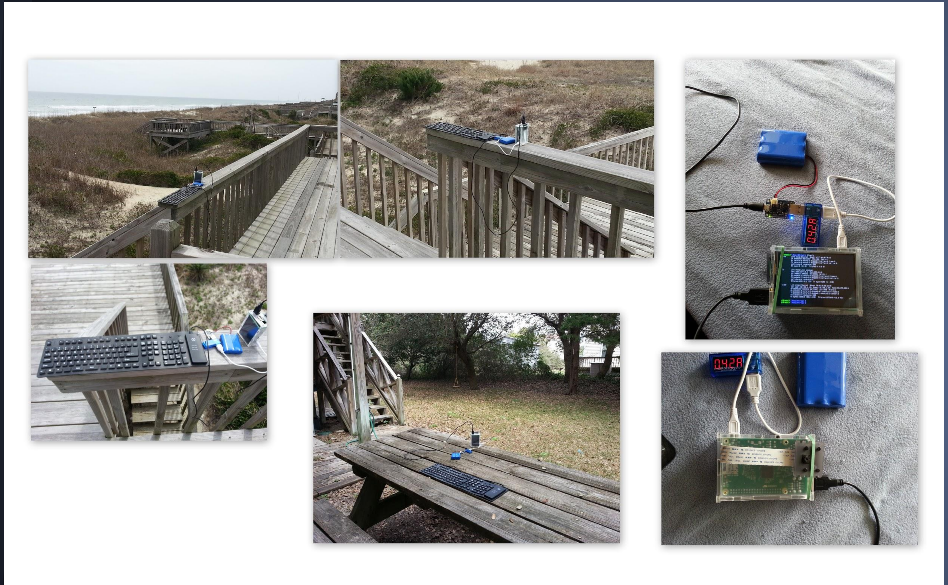
Detect, Merge and Process multiple images into a single larger consistent image.

The panorama takes the combination of techniques we learned for blending and finding features in images to produce a new expanded view for a location.

The combining of several related images together using blending and key points to merge the image produces the final result you see at the bottom from the five images above.

Assignment #9: HDR

Build an HDR hardware platform.



This platform is ideal for mobility and offers a great programmable platform for Computational Photography with Python, OpenCV, NumPy and SciPy.

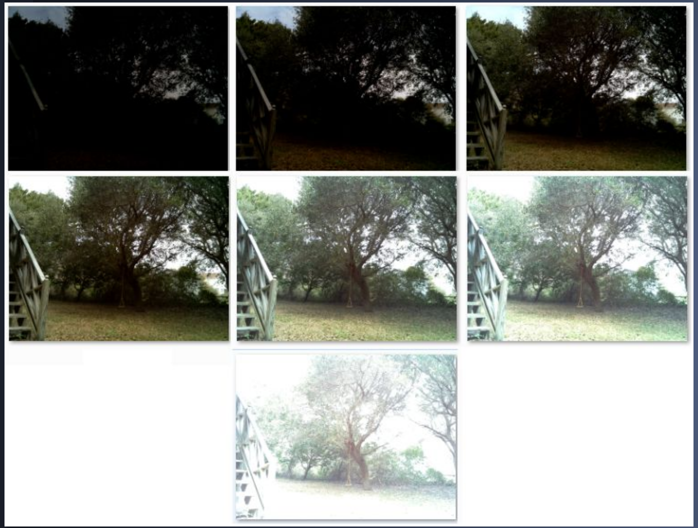
Building a HDR hardware platform

Raspberry Pi 2 with a SLI 5mp Pi Camera, a 3.5 inch PiTFT screen and WiFi makes interactions easier while mobile. The addition of a flexible membrane USB keyboard allows for input when necessary.

A blue three cell 3.7V 6600mAh lithium ion battery pack attached to a PowerBoost 1000c USB 5V pass-thru charger give portable power that can be plugged into an AC adapter.

Assignment #9: HDR

Use epsilon, merging, and pixel processing to produce a merged high dynamic range (HDR) image.



Here are seven images from my backyard at the beach as an example of a wide range of brightness. You can see a progression from the standard EV -4 up to EV +2.

The final HDR image has a greater range of luminance levels made possible by combining several different narrower range exposures of the same subject matter from the left.

Assignment #10: Pictures of Space: Panorama

An interactive 2D space.

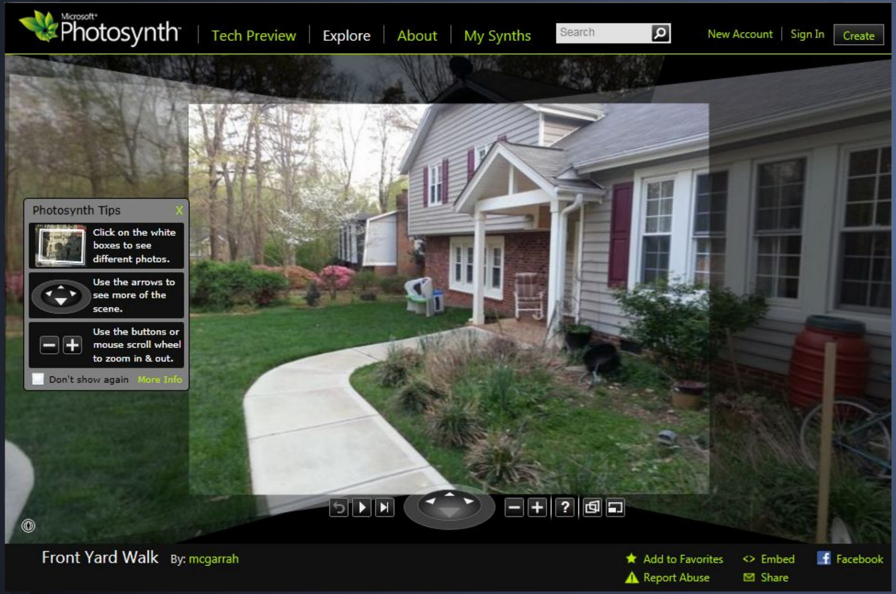
Panorama 1: “Front of House”, 21/13 Pictures. Microsoft ICE & Photosynth.net

Panorama 2: “Walk on the Farm, 7/5 Pictures. Microsoft ICE & Photosynth.net

A panorama expands the field of view as above 360° for the top and 180° for the bottom. This is an extension of the prior panorama work but included a viewer for viewing as if standing in the location.

Assignment #10: Pictures of Space: Photosynth

An interactive 3D space.



"Walk to Front Door",
Synthy 100%
Photos 197

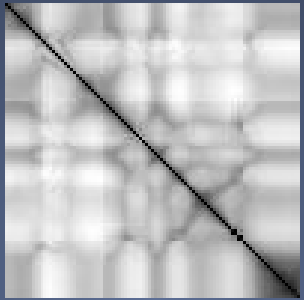
This is a PhotoSynth which is a created with a collection of combined 2D image to produce a 3D representation of the space. The difference from a Panorama is that movement through the 3D space is possible.

Click the image to experience this 3D view.

The Microsoft Photosynth web-based software was used to produce this representation. A point cloud of relationships between the individual 2D photos is produced and this will be discussed further in the Final Project section.

Assignment #11: Video Textures

Detecting differences in video frames and finding commonality for loops.



Diff Matrix

Video textures find repeatable sequences in a set of video frames. This allows us to create infinitely long video sequences or commonly called GIF loops. A difference matrix is used to find the best looping locations with darker being most similar.

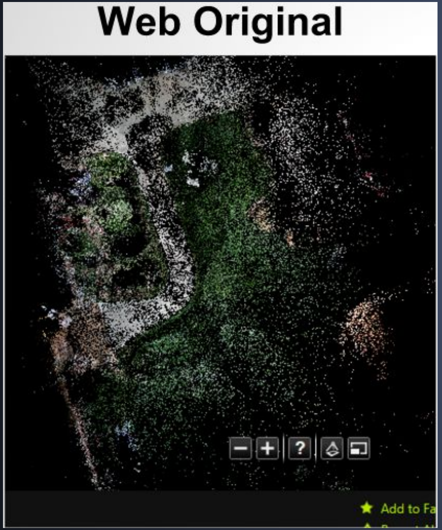
Final Project: Goal and Motivation

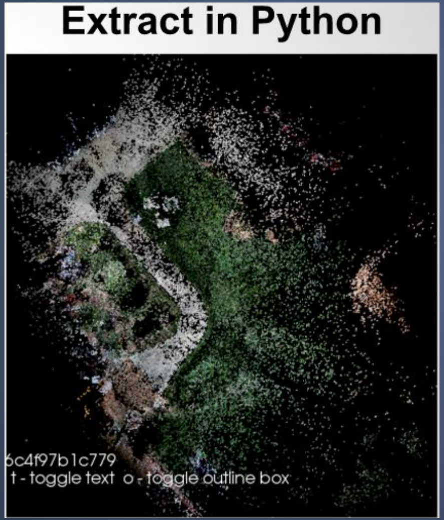
The goal of my project is to have full access to the Photosynth point cloud data for programmatic evaluation in a fully open environment. Additionally, the data is to be exported into Wavefront OBJ and Stanford PLY (Polygon File Format) which can be used in MeshLabs and other tools.

My initial work was an attempt to replicate the Photosynth processing locally and led to a couple of software packages. The first was the "VisualSFM: A Visual Structure from Motion System" that allowed for generating a point cloud from a series of photos. The second was the "MeshLab extensible mesh processing system" which allowed me to perform operations on the point cloud data. These included visualizing the point cloud and creating surfaces and meshes. This was successful in generating a lower quality pipeline for point clouds at high computational cost.

My interest in 3D spaces and their relationship to photo, shapes, meshes and derived objects began during the class assignment on Photosynth. The lack of access to the raw data was frustrating and my choice to tackle this problem came from that frustration. The Python platform and some web service and visualization libraries offer a fertile work space for learning more about 3D spaces.

Final Project





To the left is the Microsoft Photosynth close source web application interface with an overhead point cloud displayed.

To the right is the final project code that has downloaded, extracted and visualized the same data from the Photosynth project. Additionally, the data is now exported into several open formats making the data available for use in such packages as MeshLab, Wings3D, Point Cloud Library (PCL), and other software.

Final Project Details

For more detail:

Online	Final Project Report
Demo	Video Demo
GitHub	Project Code

Thank you for the awesome course

I would like to thank Professor Ifran Essa and the TAs for a great experience.

The course material and delivery was exceptional.

I look forward to applying what I have learned in the Computer Vision course in the Fall 2016 and into my future endeavours in Machine Learning.

Thank you again.