Comp Photography (Spring 2016) Final Project

Sadeq Zabihi
zabihi@gatech.edu

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

PyMeter

An effort to compute distance of an object from the camera

Compute distance of an object

This is an effort within image processing and computer vision space to compute distance of an object from the camera. I'm implementing triangle similarity theorem using Python and OpenCV.

Motivation:
The idea came to my mind when I was hiking Torrey Pines on a weekend and when I got to the top I wanted to estimate how far I parked by looking at the parking area.

Showcase what you did. This could be many images, but this single slide should be a good pictorial of your work

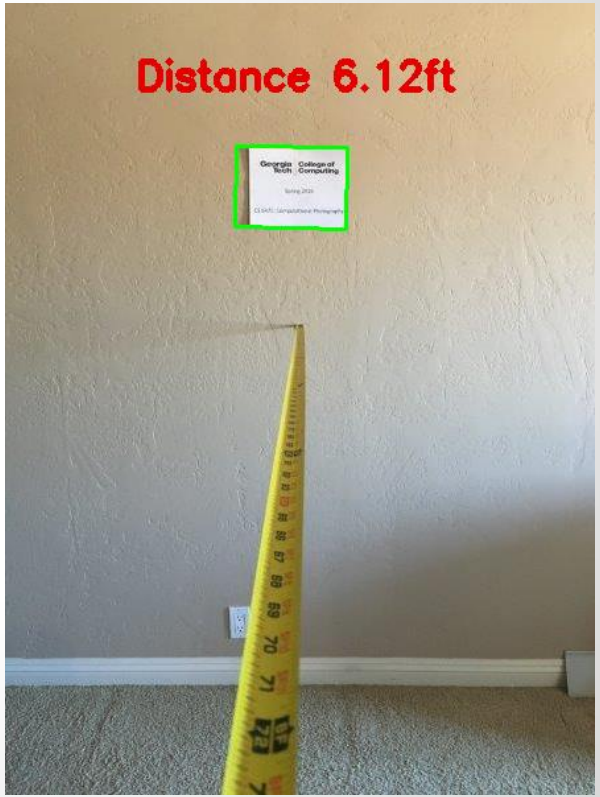
Input

Output

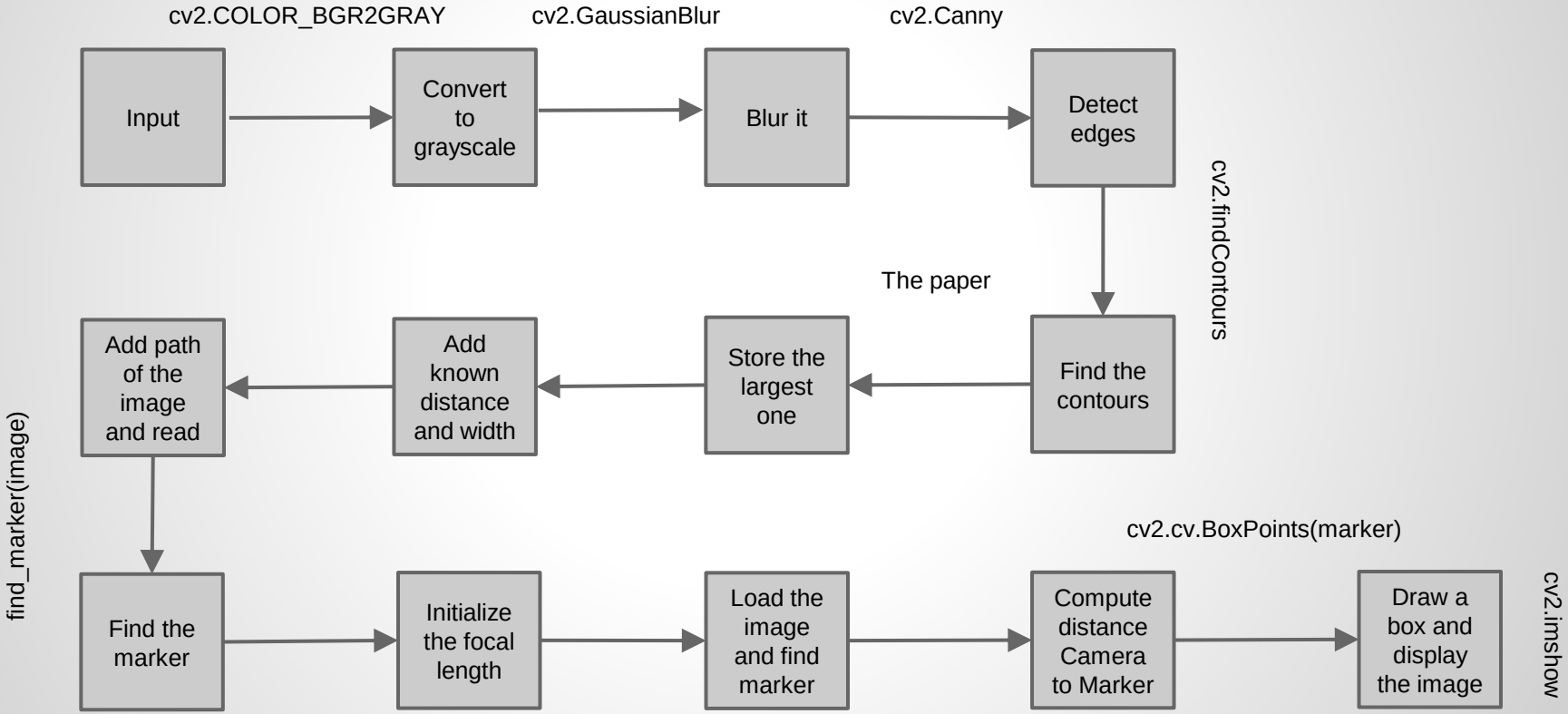
Showcase what you did. This could be many images, but this single slide should be a good pictorial of your work

Input

Output



Showcase your pipeline



What is the best way to see your project?

Inputs, outputs, screenshots, code and other items are available here:

https://github.com/msz10/CPV

What worked

I was able to use many openCV functionalities that I learned in this class including gaussian filters and canny edge detection for this project.

Using triangle simiarity I was able to determine the distance of a known object to the camera.

Triange similarity

Definitions:

- Object with a known width **W**
- Marker some distance ***D*** from the camera
- Measure the width of the object's picture in pixels **P**
- Derive the perceived focal length ***F*** of the camera

F = (P x D) / W

Triangle similarity to determine the distance of the object to the camera:

D' = (W x F) / P

Triange similarity

Examples:

F = (P x D) / W

Marker: US Letter size paper: 8.5 h x 11 w

Distance: Inches in front of the camera: D = 24

Width: Perceived width of the paper in the image is: P = 249 px

Focal length F:

F = (248px x 24in) / 11in = 543.45

Triange similarity

Examples:

D’ = (W x F) / P

D’: Change camera distance to 3 ft. (36 inches) from the marker

Width: Perceived width of the piece of paper is: P = 170 px

D’ = (11in x 543.45) / 170 = 35in

What did not work? Why?

I implemented this project using Python 2.7 and OpenCV 2.4.9. In future I’d like to implmenet it using the latest Python and OpenCV versions.

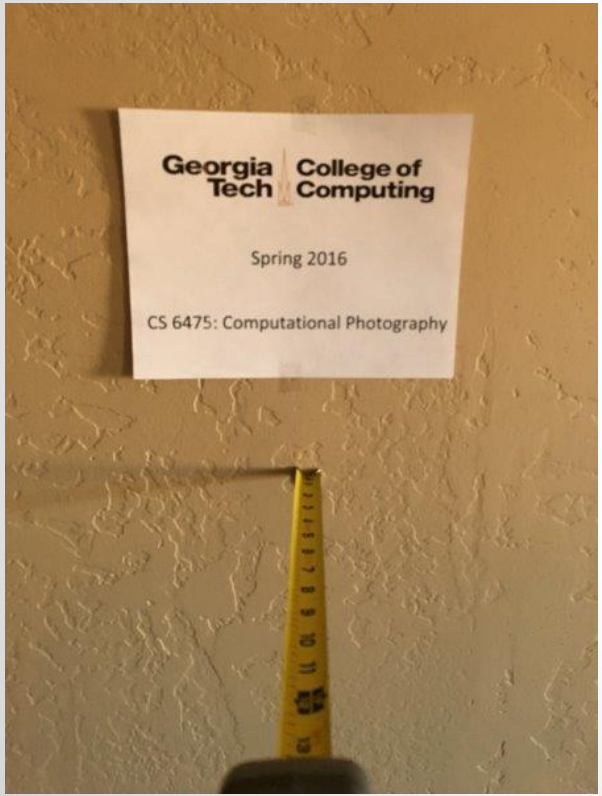
What did not work? Why?

When there was short distance between the camera and the object, I got less accurate results as well as incorrect detection of the marker. That can be resolved possibly by using a different shape/color marker (see next slide).

I think the reason it didn’t give me an accurrate result was due to the change of known distance value.

Showcase what you did. This could be many images, but this single slide should be a good pictorial of your work

Input



Output

Any additional details?

With a quick Google search I found the EasyMeasure app for iPhone and Smart Measure for android.

I ran the very same expriment using EasyMeasure but found it very challenging to get an accurate measurement.

By moving the camera even for a single inch it can change the result to 100’

Any additional details?

I ran the very same expriment using EasyMeasure app but found it very challenging to get an accurate measurement.

By moving the camera even for a single inch; it changed the result drastically.

Next slide I have two photos that shows these results. I moved the camera an inch forward and the distance calculation changed from 200’ to 300’.

Showcase what you did. This could be many images, but this single slide should be a good pictorial of your work

Output1

Output2

Showcase what you did. This could be many images, but this single slide should be a good pictorial of your work

Output

In this photo the camera is only ~ 6ft (72’) away but EasyMeasure shows the distance 12.5ft (150’).

Any additional details?

Trangle similarity requires the user to know the width of the object as well as the distance of the marker for the first step of this experiment.

Additional work is required on my side to limit the pre-requisits of this expriment. The ideal scenario is an app that can show you distance of a marker live.

References / Pointers

https://itunes.apple.com/us/app/easymeasure-measure-your-camera!/id349530105?mt=8
http://answers.opencv.org/question/24096/opencv-camera-object-distance-with-single-camera/
http://answers.opencv.org/question/5188/measure-distance-from-detected-object-using-opencv/
http://photo.stackexchange.com/questions/12434/how-do-i-calculate-the-distance-of-an-object-in-a-photo
http://blog.tibarazmi.com/measure-the-distance-of-object-to-the-camera/
http://www.samontab.com/web/2014/06/installing-opencv-2-4-9-in-ubuntu-14-04-lts/
http://opencvpython.blogspot.com/2012/06/contours-4-ultimate.html
http://stackoverflow.com/questions/14038002/opencv-how-to-calculate-distance-between-camera-and-object-using-image
http://math.stackexchange.com/questions/1606047/calculate-distance-to-person-from-camera-if-camera-is-on-the-floor
http://www.lnse.org/papers/80-VC3006.pdf
http://dsc.ijs.si/files/papers/S101%20Mrovlje.pdf
http://www.regentsprep.org/regents/math/geometry/gp11/LsimilarProof.htm
https://www.khanacademy.org/math/algebra-basics/core-algebra-geometry/copy-of-triangle_similarlity/v/similarity-postulates