iMeasurer

Computer Vision (CSC 767) Fall ‘14 - Final Project

Rongzhong Li, Saurabh Morchhale, Xin Zhou

ABSTRACT

Image ruler has been used for measuring the height of different objects in different ways. In this project a new measurement algorithm is presented which generates dimensional measurements for different sizes of objects, such as shoes, human, buildings, and etc. In total, two images and the known moving distance to take these two images are required to implement the algorithm. The method depends on results from the two similar projective triangles geometry. So to get a good estimation, the planes of the object and the phone have to be parallel. The accuracy of the algorithms depends on accurately selecting of one pixel point from top and bottom of the object, respectively, and the window size around the selected two pixel points.

1. INTRODUCTION

When looking at photos, people can roughly intercept the image and get its length information. However, the image taken by a camera is a 2D projection on the camera sensor. It may produce some illusions when looking at the scene from only one perspective.

A large object in a photo can be either large by itself, or too close to the camera.

There are quite a lot computer vision algorithms dealing with reality reconstruction[1]. However they are too computationally expensive for our simple purpose. We want a algorithm that's easy to implement and fast to calculate.
Intuitively if we have two photos taken from different distances, we can get a better understanding of the truth. In the sense of algebra, we can eliminate one degree of uncertainty by introducing another equation.

2. METHODOLOGY

2.1 Algorithms

To solve the problem mathematically, we first plot the ray chart of a camera.

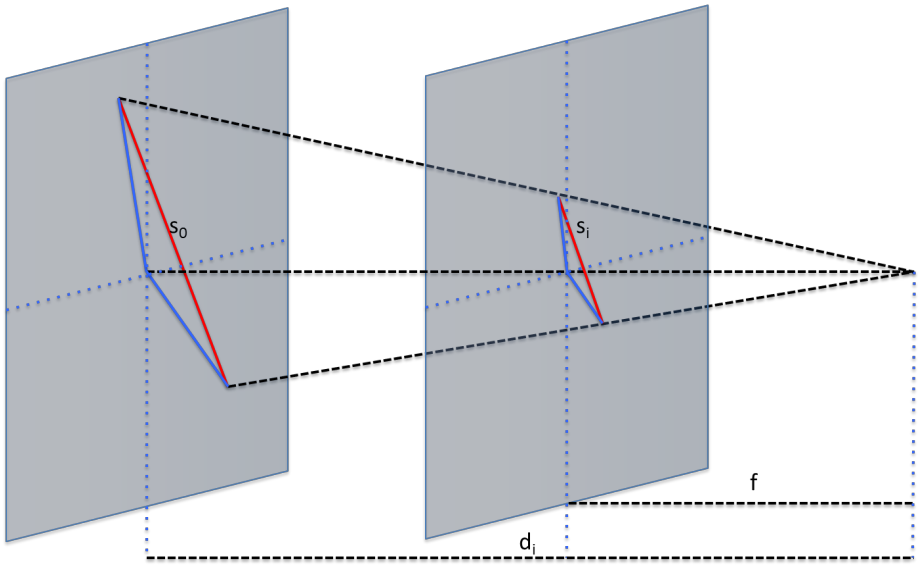


Figure 1. Cartoon of the ray line in camera.
s_0 is the real length on a distant plane. s_i is the projected length on the sensor plane. the distance between the lens and s_0 is d_i, while the focal length(distance between lens and sensor) is f. These four lengths form a set of similar triangles. A traditional measurement method requires the distance d_i to figure out the real length. The problem is the object may be too far from the view point or even inaccessible, like a distant building over the river. And when you can't go there, you can't have an accurate estimation.
We can use another approach to solve the equation. The original equation is linear equation with one unknown. Since the distance is also unknown, we need another equation to solve two unknowns. We use images at two distances while the relation between the distances can be determined.

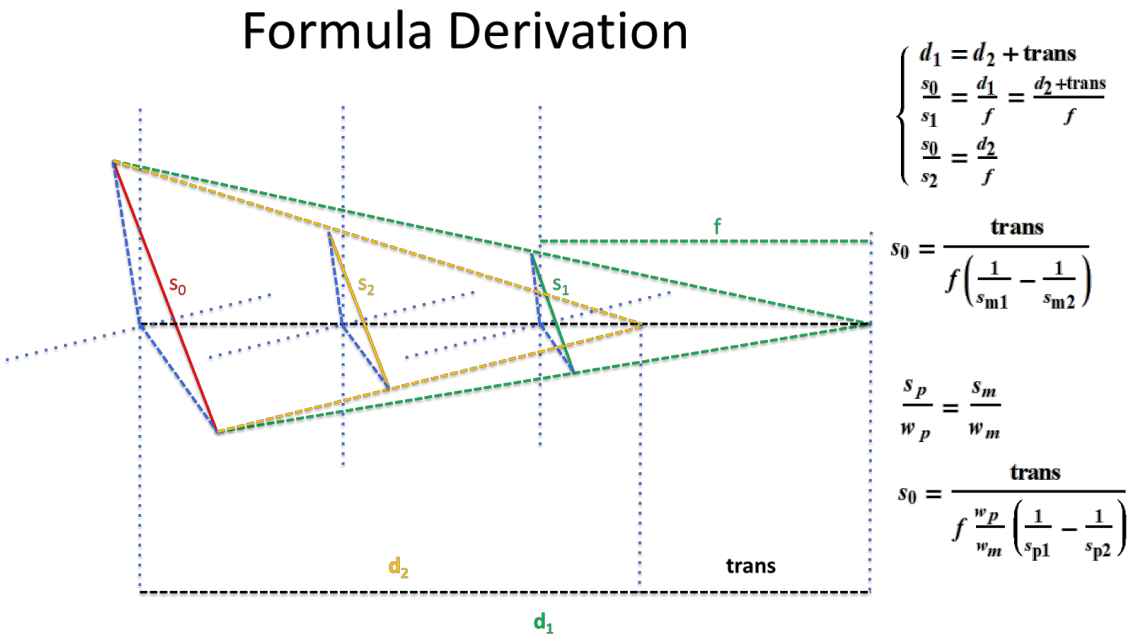


Figure 2. Key algorithm.
We take two photos of s_0 at d_2 and d_1. The distance between d_2 and d_1 is *trans*. The red, yellow and green lines form two pair of similar triangles. By solving the geometry equations, we can calculate the real s_0.

The accuracy of this method depends on accurately measuring the translation distance and picking the projection's edge points. And according to the formula, the translation has to be large to make the two projection very different so that $(1/s_1\text{-}1/s_2)$ is not close to 0. We can get the translation using some known dimensions, like A4 paper, human height, ruler, or even the length of the used phone(with camera) itself. Till now, the key to solve the real s_0 is the objects' dimensional measurement in pixels of the two images.

2.2 Image pre-processing and key points selection

As discussed before, now to calculate objects' dimensional measurements (for example, height for human) in pixels of two images are needed, but it causes two problems: firstly, it's not efficient to calculate the height in pixels just from two points; secondly, it's difficult to choose exactly the same two points from the two images. To solve these problems, a canny edge detection[2] is used to make objects' edge features pop out, and a window size can be changed to restrict the edge point we picked to a reasonable window. Furthermore, for all the edge points in the selected window, a SIFT algorithm and homography transform[3] can be implemented to find multiple matched edge points in the two images, and then we can calculate all the possible heights in pixels.

3. RESULTS

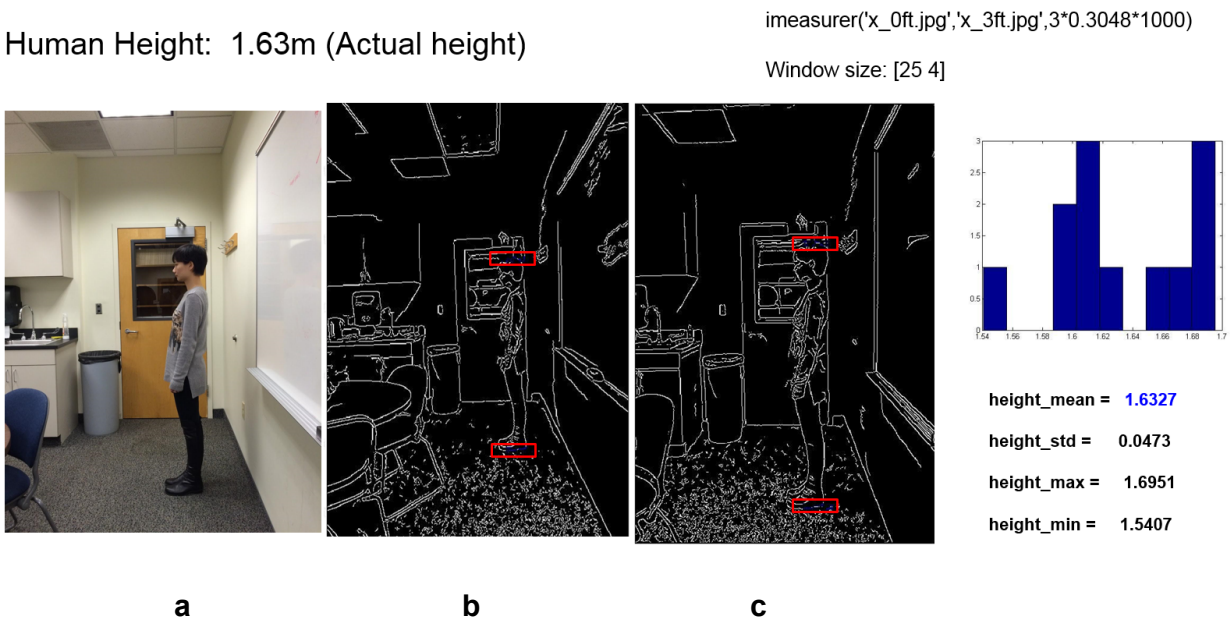


Figure 3. Human Height Estimation using two images with moving distance around 1m. Histogram figure shows the distribution of all possible measurements.
a, d) are Original pictures. b,c,e,f) are Canny Edge pictures

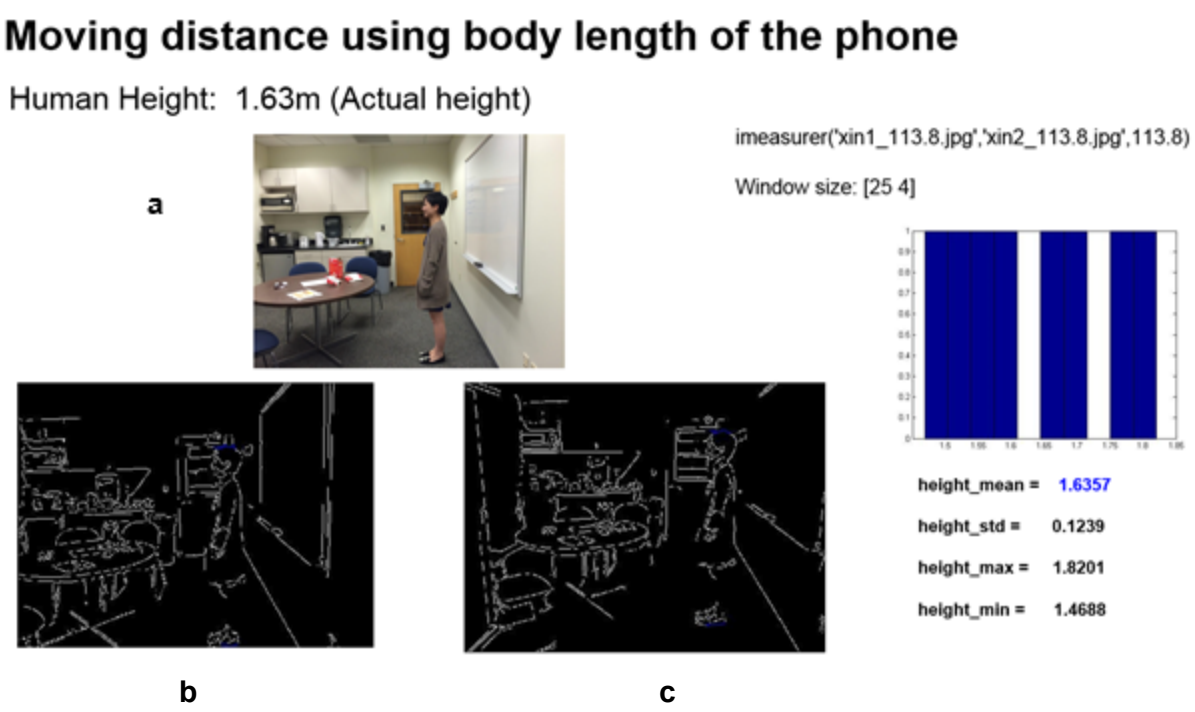


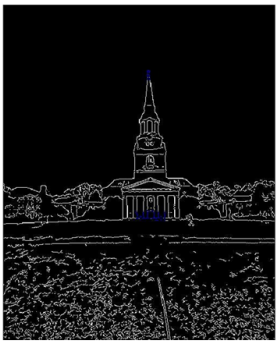
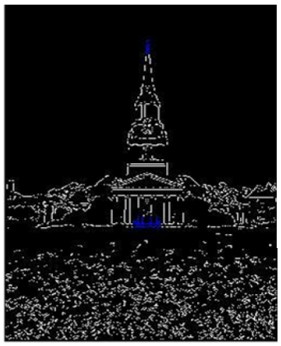
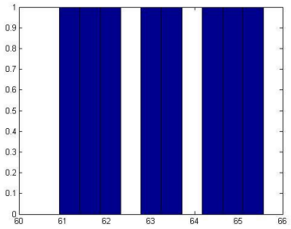
Figure 4. Human Height and Object dimension estimation using two images with moving distance of body length of the phone. Histogram figure shows the distribution of all possible measurements. a, d) Original pictures. b,c,e,f) Canny Edge pictures.

Moving distance around 23m

Building Height: 66m (Actual height)

imeasurer('a1_0ft.jpg','a1_75ft.jpg',22.86*1000)

Window size: [25 14]

height_mean = 63.2935

height_std = 1.6194

height_max = 65.5654

height_min = 60.9171

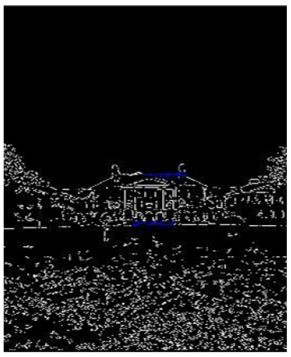
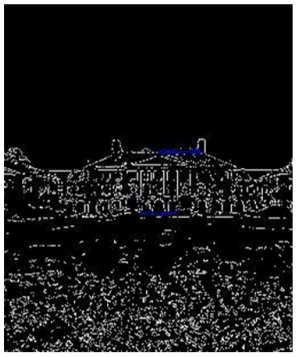
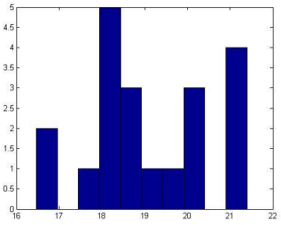
a **b** **c**

Moving distance around 23m

Building Height: 16m (Actual height)

imeasurer('b1_0ft.jpg','b1_75ft.jpg',22.86*1000)

Window size: [35 4]

height_mean = 19.1852

height_std = 1.4769

height_max = 21.3951

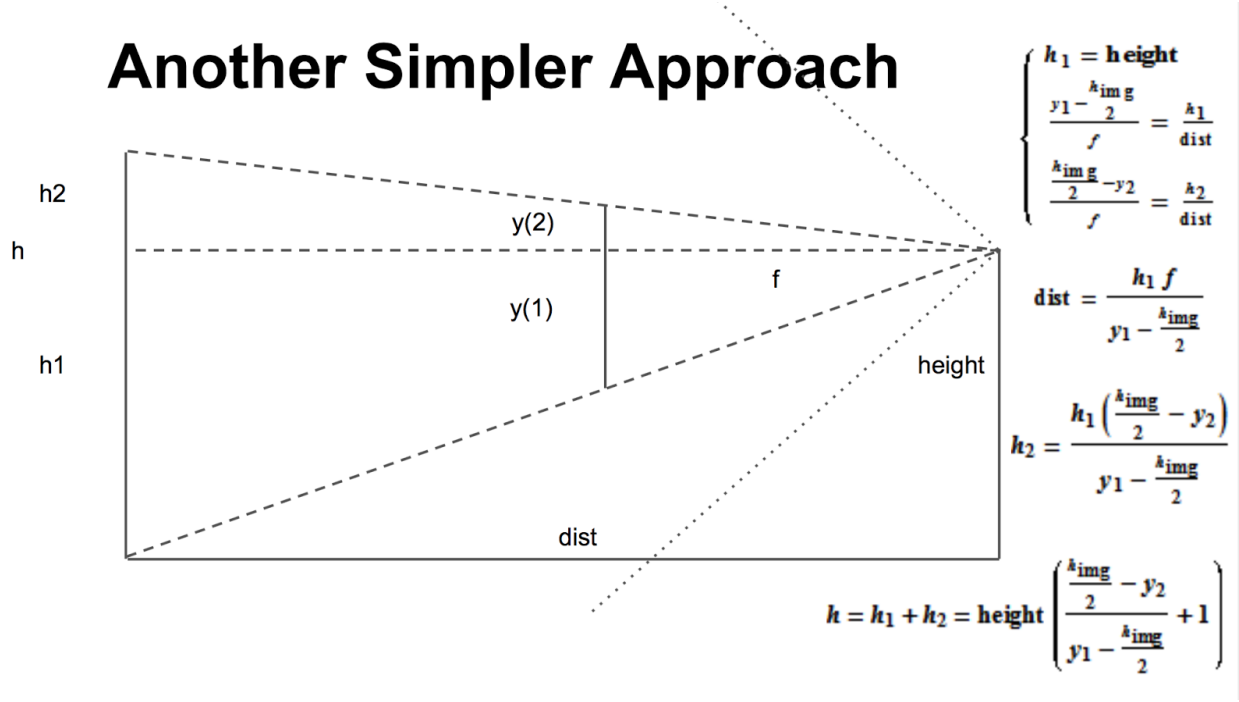
height_min = 16.4646

d **e** **f**

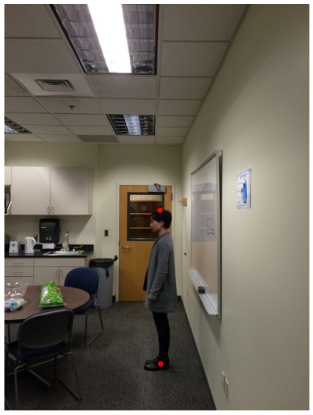
Figure 5. Building height estimation using two images with moving distance around 23m. Histogram figure shows the distribution of all possible measurements. a, d) Original pictures. b,c,e,f) Canny Edge pictures

3. Another simple approach

While developing the previous method, we kept looking for existing methods for achieving our purpose. And one ios app is doing things very similar to our project[4]. After understanding its method, we simplified its procedures and developed another approach for measuring the height in an image. The mechanism is demonstrated as below.



Human Height: 1.63m (Actual height)



simple('1.7m_x.jpg',1700)

Height of the photographer (reference height) : 1.7m

Real_h = 1.6637m

Figure 6. Human height estimation using simple algorithm which uses the photographer's height as reference.

References

1. Measuring Planar Objects with a Calibrated Camera
   http://www.mathworks.com/help/vision/examples/measuring-planar-objects-with-a-calibrated-camera.html
2. Canny Edge Detector:
   https://github.com/PahanPerera/EdgeDetector/blob/master/canny.m

3. SIFT
   http://www.scholarpedia.org/article/Scale_Invariant_Feature_Transform

4. EasyMeasure
   https://itunes.apple.com/us/app/easymeasure-measure-your-camera!/id349530105?mt=8