Indoor Localization with Computer Vision

INTRODUCTION

Due to lack of the GPS signal in the building, accurate indoor localization has been considered an important application among the computing research community. Multiple methodologies are used to achieve the goal, such as tracking a user's path by embedded sensor of a mobile phone, localizing a user's position based on multiple Bluetooth signal strength, and using the camera to locate a user's position in the map, etc. In this project, we present an indoor localization application using the principles of computer vision.

The primary goal of our project is to detect multiple specified objects, and localize the position of a camera or a user in the map based on these objects' position. But due to the time limitation, our simplified setup is such that localizing the position of the camera by detecting a simple object with known shape and size; in the experiment, a Rubik's cube of 5.7 cm was used as the target. For imaging an iPhone 6 plus is used, whose focal length for still camera is 4.15 mm and 4.6 mm for the video camera. The pixel size in both still image and video is $1.5\,\mu m$. For the video footage, the frames of the video are extracted and then used as single images for the calculations.

POSSIBLE METHODOLOGY

There are four possible methodologies for localizing the objects, which are as follows:

- **Perspective**: If the camera is in the vertical position, then it is easy to find the distance between the camera and the vertical edge by using perspective projection with known dimensions of the object and the focal length. The challenge is to locate the camera by perspective when the camera is at an angle.
- **Extrinsic parameters:** To calculate the extrinsic and intrinsic parameters at least six points are needed, preferably more for accurate results. We, however, would like to calculate the extrinsic parameters with fewer than six, given the intrinsic parameters derived from the camera calibration.
- **Stereo camera setup:** The disparity can also be used to get the distance between camera and object. The problem with stereo camera setup, however, is that only one camera was used and using two different images from different positions would not provide a result, as the baseline between the images is unknown.
- **Optical flow:** Optical flow is a technique which focuses on finding the difference between multiple frames. It's a good technique for detecting a moving object with a fixed camera. And it's possible to get the difference between frames with a fixed object if we know the camera's angle in each frame(always parallel in practical usage). But in our situation, the angle or the baseline between each frame is unknown based on irregular human movement.

IMPLEMENTATION

To achieve the goal, the project is divided into two parts: object detection and localization. To detect the object, the first step is color detection of a known object (cube) with selection of one face. After detecting the selected face, the corners of the cube are detected, which are used in a further localization procedure. The localization procedure uses the perspective projection and trigonometric functions to obtain the the x,y coordinates of the camera based on the corners of the detected object.

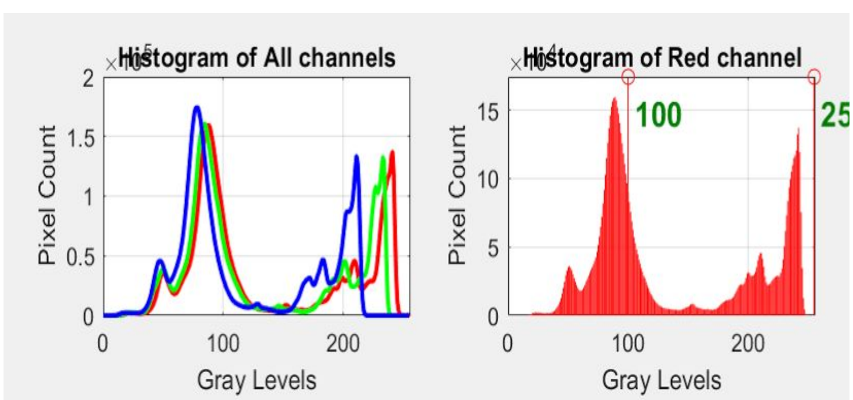
OBJECT DETECTION

With the Rubik's cube width of 5.7cm edges as mentioned before, the object in the captured image or video is detected in the following steps:

- **Color Detection:**
  - The Red-Green-Blue channels are extracted first from the image.

Figure.1: The RED-GREEN-BLUE channels of the image

  - Then the histograms for the above RGB channels are generated.



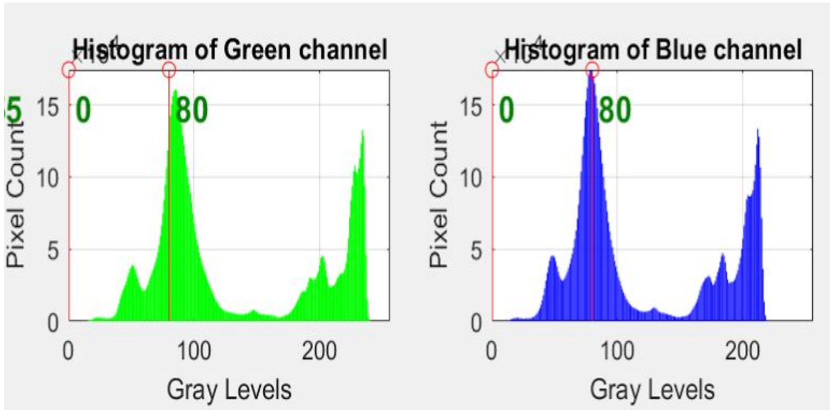


Figure.2: Histograms for the red, green and blue channel

- As the red face of the cube is taken into consideration, we evaluate the threshold value to keep majority red channel, and cut off the most contents of blue and green channel based on the histogram. By introducing these threshold values we form the masks of each RGB channel of the captured cube. Figure.3 shows the mask images of is-red, not-green, and not-blue channels:

Figure.3: The RGB masks based on “red-only thresholds”

- The masks of all the colors are integrated together to produce the mask of the red ONLY object mask.

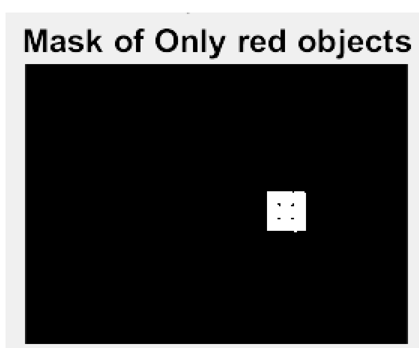


Figure.4: Red-only mask by integrating is-red mask, not-blue mask and not-green mask

- After creating the mask of the red only object, to optimize the detected object,very small objects(<5000 pixels) are removed to isolate the required object. The voids in the image are filled to produce better results.

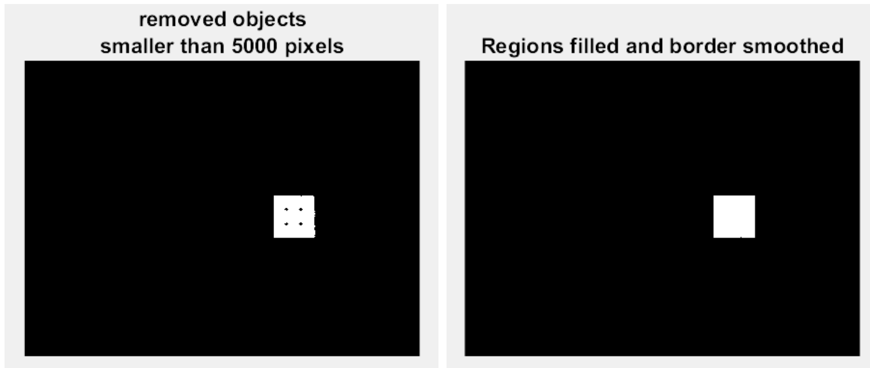


Figure.5: Mask of the red object after removing the very small objects and filling the voids.

- A gaussian filter is applied to smooth the edges and curves of the object. The sigma value used was 20.0. When applying a smaller value <10.0, the edges were not smoothed well and a high sigma value caused the shape of the cuboid to not be maintained and the corners were not identified.

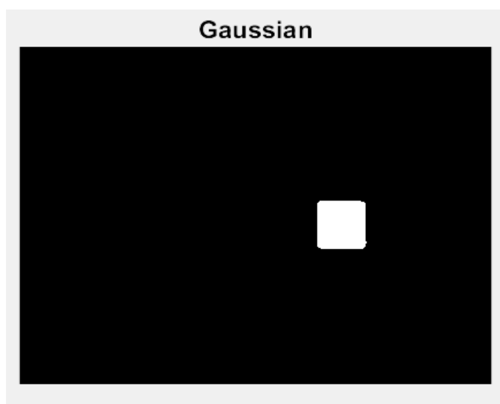


Figure.6: The object after applying the gaussian filter.

❖ **Corner Detection:**

- The Sobel edge detector was used to get the edges of the object.

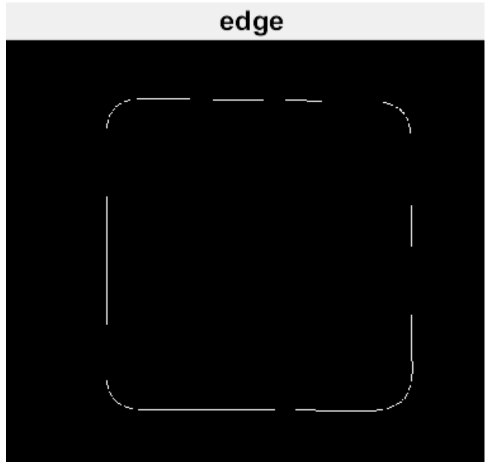


Figure.7: The edges of the object we get using the Sobel edge detector.

- The cartesian coordinates of the edge are converted to the polar coordinates and the max peaks of the edge polar coordinates are identified as corners. A mean filter was also applied to get a better result.

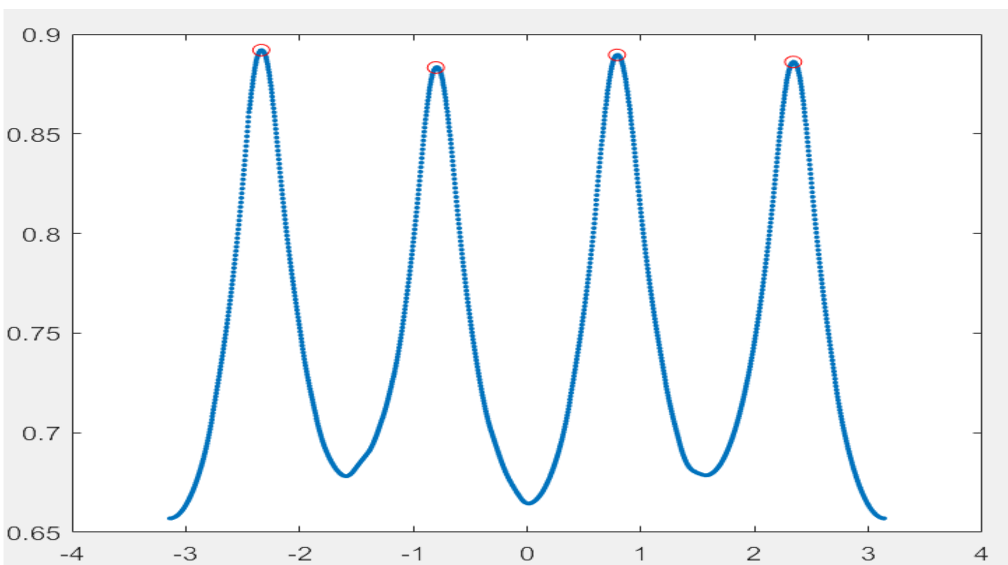


Figure.8: The polar coordinates of the edges. The max peaks (red circles) are corners.

- From the polar coordinates, the corners' coordinates in image were obtained, and these coordinates are used for the localization of the object.

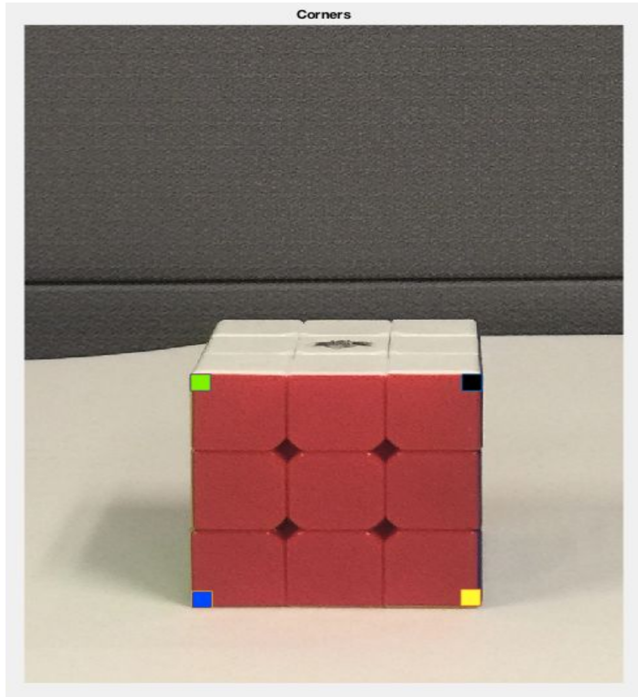


Figure.9: The corners of the cube's red face are detected.

LOCALIZATION

Based on detected corner of the objects, we use the perspective projection to find the x,y coordinates of the camera. Figure.10 shows the geometry of calculating the distance between vertical edge of the object and camera.

To simplify the challenge, we made an assumption: the camera is always kept in the vertical position. In our experiment, we assign the lower left corner of the cube to be the origin in the world space.

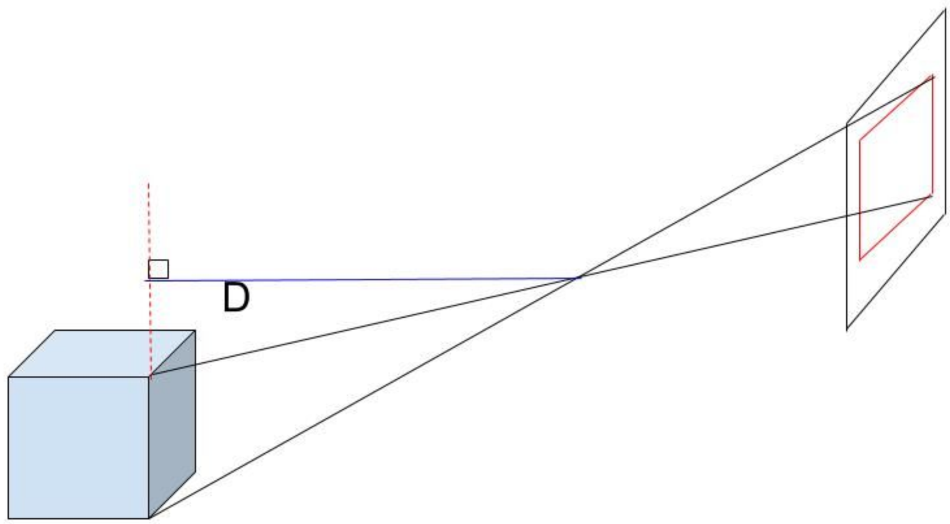


Figure.10: Perspective of vertical camera

From figure.10, the distance between the selected vertical edge of the cube and the camera is:

$$D = f * \frac{Cube\ Length}{Cube's\ vertical\ edge\ in\ pixels}$$

The cube's length is 5.7 cm and the pixel difference of the vertical edge is calculated by the corners that were detected from last section. As we found four corners of one selected face of the cube, two of the vertical edge could be detected in the image, and we can get two different distances between the camera to the two vertical edges. Figure.11 shows the triangulation which is formed by the two distance and the cube edge.

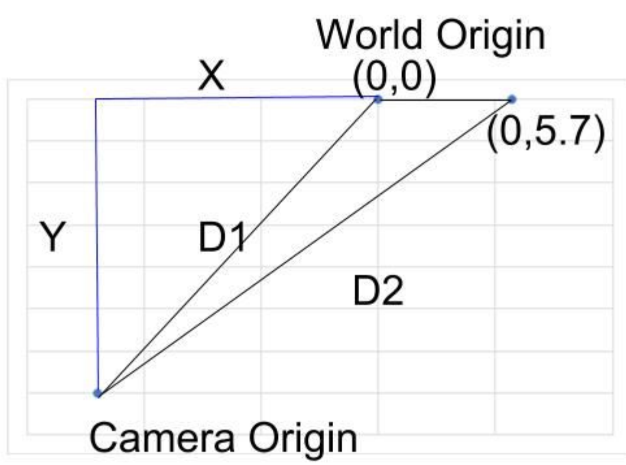


Figure.11: Triangle constructed with distances between two vertical edges and cube length

From figure.11 above, with three known edges of triangulation, Heron's formula can be used to solve for the x and y coordinates of the camera.

$$l = \frac{(D1+D2+Cs)}{2}$$

$$S = \sqrt{l(l-D1)(l-D2)(l-Cs)}$$

$$y = \frac{2*S}{Cs}$$

Where l is half of the perimeter of the triangulation, D1 is the distance between camera and first vertical edge of the cube, D2 is the distance between camera and second vertical edge of the cube, Cs is cube size, y is y coordinate of the camera position.

The x coordinate can then be solved by the pythagorean equation are we know the value of y and D1, creating a right triangle.

$$x = \pm\sqrt{Cs^2 - y^2}$$

Finally the direction of the x-coordinate is calculated by the angle of the triangle, θ, which is constructed by D1 and cube horizontal edge:

$$\cos\theta = \frac{D1^2 + Cs^2 - D2^2}{2*D1*Cs}$$

If $\cos\theta$ is less than zero, then the direction of the x coordinate is also less than zero and vice versa.

RESULTS

Using the algorithm described in the previous pages, a video was taken with the cube always in frame and the camera held vertically. The full path traced is generated as below:

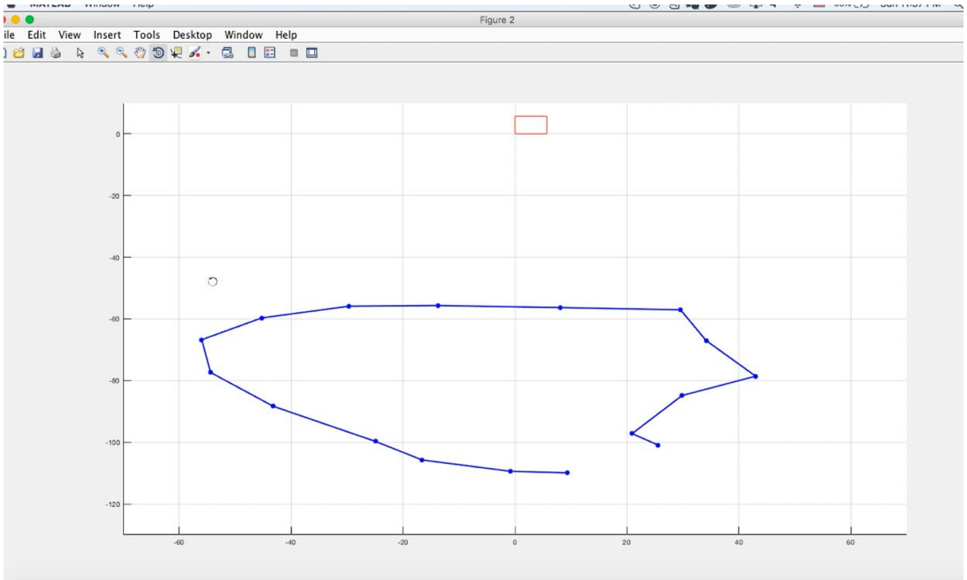


Figure.12: The full path of the camera during filming

The camera begins in the center, moves left, and returns close to its original starting position.

CHALLENGE

The simplified setup returned accurate results but for further work there are some challenges. In the simplification, the camera is at a vertical position for every image or throughout the video; there are some errors that are encountered when the taken images are at angle.

To get perspective with a rotated camera, in theory, we can use only 6 points to get both extrinsic and intrinsic parameters from a single image, and it can solve all our challenges because translation matrix includes the information of the camera's position. But in practice, it's really inaccurate using only 6 points, because of the error and noises encountered when we select the points from the image, irrespective of the manual or automatic detection by the program. On the other hand, the automatic object and corner detection can get at most 7 corners from a cube. Another way tried was reversely using the camera parameter equations and solving for [R|t] with known intrinsic parameter:

$$\begin{bmatrix}u\\v\\1\end{bmatrix}=\frac{1}{Z}K[R\,t]\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

$$\frac{[R\,t]}{Z}=K^{-1}\begin{bmatrix}u\\v\\1\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}^{-1}$$

The [R| t] matrix calculated from this equation returns normalized values of the rotation and translation vectors as the scalar component Z is unknown. To get accurate values of the camera parameters a minimum of six points are needed. More points can be introduced as the object's size is known but if all the points are in one plane the results obtained are not accurate.

A possible methodology is that the face of the cube is known as a square, it's possible to get a rotation matrix by extending the vanishing point to infinity. The rotation matrix could be used to solve the normalized [R|t] matrix above to get the exact translation matrix.

CONCLUSION

We presented an indoor localization methodology by using camera and specified object with object detection and perspective projection. The result is accurate with vertical camera setup, and also could be used for both captured images or videos. Future work is to focus on multi object detection with unknown shape and size, and also to localize the objects with the camera at any angle.