3D LOCALIZATION OF OBJECTS USING MULTIPLE CAMERAS

GARIMA SINGH
Department of Electrical Engineering
Indian Institute of Technology (BHU)
Varanasi-221005
garima.singh.eee20@itbhu.ac.in

GEEREDDY SATHWIKA REDDY
Department of Electrical Engineering
Indian Institute of Technology (BHU)
Varanasi-221005
geereddy.sathwikar.eee20@itbhu.ac.in

ISHWAR LAL KUMAWAT
Department of Electrical Engineering
Indian Institute of Technology (BHU)
Varanasi-221005
ishwar.lalkumawat.eee20@itbhu.ac.in

Under the supervision of:
Dr. Sandip Ghosh
Professor
Department of Electrical Engineering
Indian Institute of Technology (BHU)
Varanasi-221005

12th of May, 2022

ABSTRACT

Using a multi-camera sensor system, we explore a mobile robot's three-dimensional (3D) localization. The calibrated and synchronized cameras are installed in predetermined locations around the area. We propose a real-time 3D localization algorithm that calculates the mobile robot's 3D location in an indoor environment by fusing the 2D image coordinates from several viewpoints synchronously in a multi-camera sensor system. Furthermore, our approach is real-time and straightforward to implement. According to experimental results, the suggested algorithm may provide reliable, efficient, and real-time 3D localization in interior contexts.

1 Introduction

A common problem in the field of autonomous robots is how to obtain the position and orientation of the robots within the environment with sufficient accuracy. Several methods have been developed to carry out this task. The localization methods can be classified into two groups : those that require sensors onboard the robots and those that incorporate sensors within the environment. To tackle this problem we have worked on a solution to incorporate sensors/cameras in virtual 3D environments and localize their position through Optical Tracking.

3D localization or 3D position tracking can be defined as the measurement of a 3D position and orientation of one or more objects or subjects that move in a defined space, relative to a known location. Optical tracking is a 3D localization technology based on monitoring a defined measurement space using two or more cameras (multiple-cameras).

Optical tracking has proved to be a valuable alternative to tracking systems based on other technologies, such as magnetic, acoustic, gyroscopic and mechanical. Optical tracking is less susceptible to noise from the environment and it does not suffer from drift problems. It also allows many objects to be tracked simultaneously.

2 Approach

We consider a multi-camera system with four cameras in it. But the proposed approach can be applied to multi-camera systems with any number of cameras. We assume the camera parameters are already known, such that the parameter for the α_{th} camera are focal length $f_\alpha = (f_\alpha^x, f_\alpha^y)$, distortion coefficient $K_1^\alpha, K_2^\alpha, K_3^\alpha, P_1^\alpha, P_2^\alpha$, position in real-world coordinates C_α.

The task can be divided in five major tasks:

1. Locating the Robot for each view.
2. Remove distortion effect.
3. Conversion from 2D pixel plane to 3D camera frame.
4. Conversion from 3D camera frame to 3D world frame.
5. Finding optimal solution.

2.1 Locating the Robot for each view

In order to locate the robot in each view we attach a red helmet or a red sphere as an artificial landmark to the robot which makes it distinguishable from the surroundings. Now to detect the centroid of the red sphere, we segment out the red colour which gives out a mask. Then we apply contour detection to the obtained mask to get contour information and then we get the center of contour in the images using computer vision techniques.

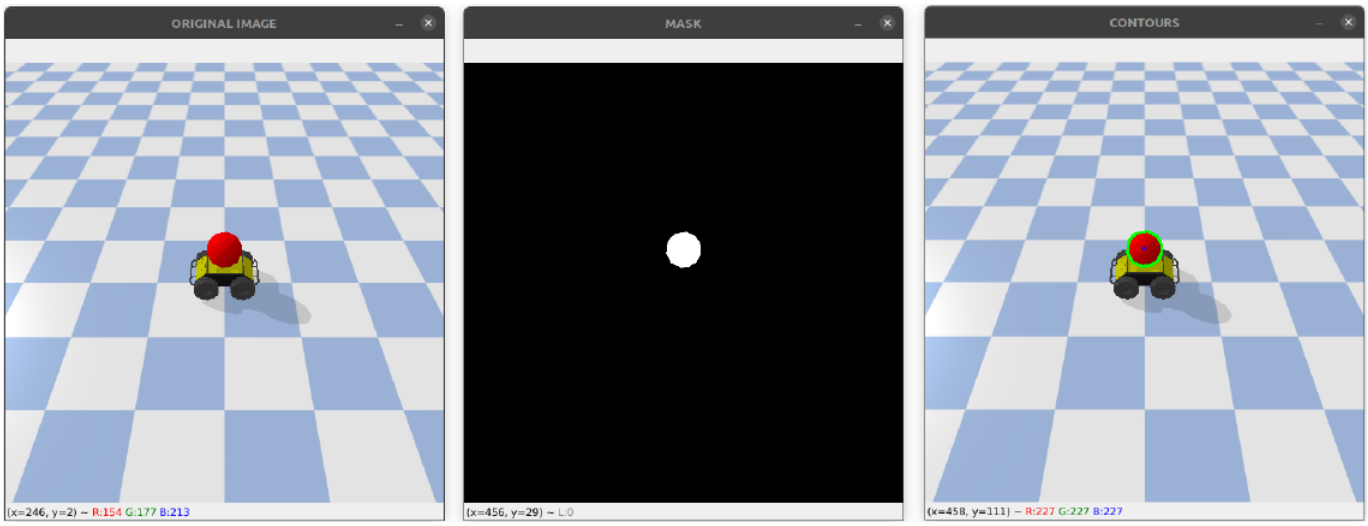

Figure 1: Masking and contour detection of an object in an image.

2.2 Remove distortion effect

If the image coordinates of the detected centroid from the α_{th} visual sensor are $p = (x_p^\alpha, y_p^\alpha)$, the normalized coordinates are (x_n^α, y_n^α). Assuming $r^2 = ((x_n^\alpha)^2 + (y_n^\alpha)^2)$ and distortion coefficients $K_1^\alpha, K_2^\alpha, K_3^\alpha, P_1^\alpha, P_2^\alpha$, the new normalized point coordinate, (x_α, y_α), with lens distortion removed is:

$$x_\alpha = (1 + K_1^\alpha r^2 + K_2^\alpha r^4 + K_3^\alpha r^6)x_n^\alpha + dx$$
$$y_\alpha = (1 + K_1^\alpha r^2 + K_2^\alpha r^4 + K_3^\alpha r^6)y_n^\alpha + dy$$

where the tangential distortion dx, dy are:

$$dx = 2P_1 x_n^\alpha y_n^\alpha + P_2(r^2 + 2(x_n^\alpha)^2)$$
$$dy = 2P_1 x_n^\alpha y_n^\alpha + P_2(r^2 + 2(y_n^\alpha)^2)$$

2.3 Conversion from 2D pixel plane to 3D camera frame

To plot the coordinates of the object, we firstly need to locate the 2D pixel plane coordinates of object from each view to 3D camera frame. To get the camera frame coordinates, project the 2D pixel plane coordinates on the image plane. Connect the camera center with the point and the intersection will give required coordinates. Figure 2 shows the projection from 2D camera coordinates to 3D camera coordinates.

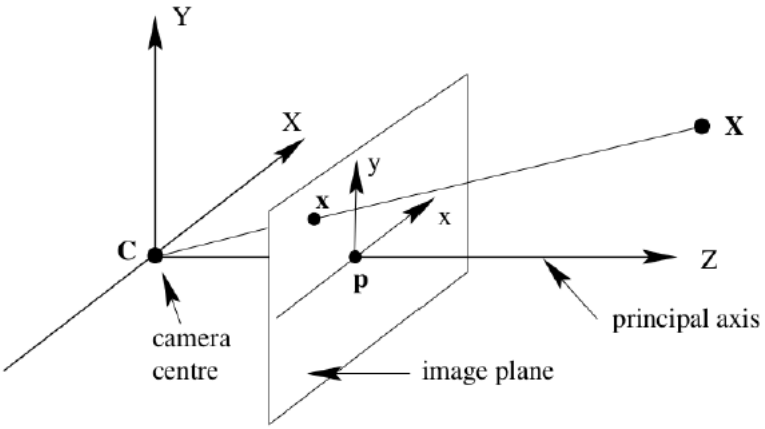

Figure 2: Conversion from 2D camera coordinates to 3D camera coordinates.

$$P_\alpha = [Z\frac{x_\alpha}{f_\alpha^x}, Z\frac{y_\alpha}{f_\alpha^x}, Z] \tag{1}$$

2.4 Conversion from 3D camera frame to 3D world frame

Now for each view, we have P_α, a 3D point in camera frame and C_α, camera position in real world coordinates which is origin in camera frame. In order to obtain 3D position of robot in real world coordinates we need to transform P_α from camera frame to real world coordinates. So we need to generate the transformation matrix to transform coordinates from camera frame to world frame. There are many methods to generate a transformation matrix given the center and orientation of the child frame with respect to the parent frame. Here we explain one such method, the Look-At transformation method.

Look-At Transformation Consider the position of the camera C_α, the point the camera is looking at L_α, and an up unit vector $\hat{U_\alpha}$ that orients the camera along the viewing direction implied by the first two parameters. All of these values are in world space coordinates. We now generate two new unit vectors $\hat{D_\alpha}$, $\hat{R_\alpha}$ where $\hat{D_\alpha}$ = L_α - C_α and $\hat{R_\alpha}$ = $\hat{U_\alpha} \times \hat{D_\alpha}$. We now generate transformation matrix as follows:

$$M_\alpha = [\hat{R_\alpha}\hat{U_\alpha}\hat{D_\alpha}C_\alpha] \tag{2}$$

where $\hat{R_\alpha}, \hat{U_\alpha}, \hat{D_\alpha}, C_\alpha$ are column vectors of length 3 and M_α is a $3x4$ matrix. We can now transform point P_α which is in camera frame to a point P_α^R which is in real-world coordinates using the equation given below:

$$P_\alpha^R = M_\alpha \begin{bmatrix} P_\alpha^T \\ 0 \end{bmatrix} \tag{3}$$

2.5 Finding optimal solution

For each view, we get a 3D line by joining the two points P_α^R and C_α. Now we obtain 4 different 3D lines joining the camera position and the 3D point. However all the 3D lines will not intersect at one point because of the errors caused

by lens distortion. Thus, there is no common intersection point of the multiple lines from multiple views, which makes it necessary to find the solution with the shortest distance to all rays.

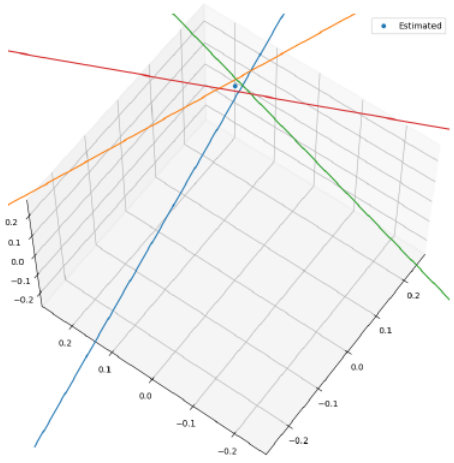

Figure 3: Four non-concurrent lines joining P_{α}^{R} and C_{α} for camera.

Considering the sum of perpendicular distances from all the four lines to a given point as the objective function L, we need to minimize this objective function to obtain an accurate solution. We use gradient descent, a first-order iterative optimization algorithm for finding a local minimum of the objective function. We perform k iterations of gradient descent to refine our solution.

$$
\begin{aligned}
L = & \perp_1 + \perp_2 + \perp_3 + \perp_4 \\
& \text{for } i = 1 : k \\
X &= X - lr \frac{\delta L}{\delta X} \\
Y &= Y - lr \frac{\delta L}{\delta Y} \\
Z &= Z - lr \frac{\delta L}{\delta Z}
\end{aligned}
$$

3 Applications

1. Surveillance of mobile objects in public spaces.
2. Used to monitor work environments.
3. Implemented in the field of entertainment and communication like digital animation.
4. In healthcare industry in the form of bio-mechanical analysis and clinical diagnosis.

4 Results

We were able to achieve the aim of localising 3D objects with the help of multiple cameras. We created a simulation consisting of a mobile robot and four cameras at the corners of a room using PyBullet. PyBullet is a Python module for robotics simulation and machine learning, with a focus on simulation-to-real transfer. We implemented the above proposed approach in our simulated world to validate the results.

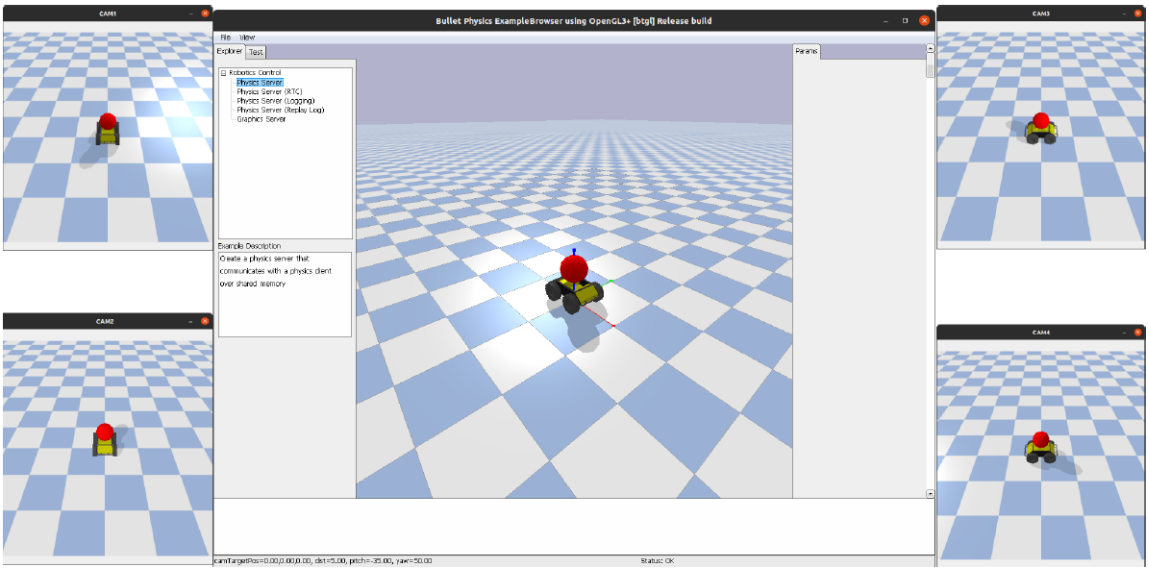

Figure 4: Simulated World.

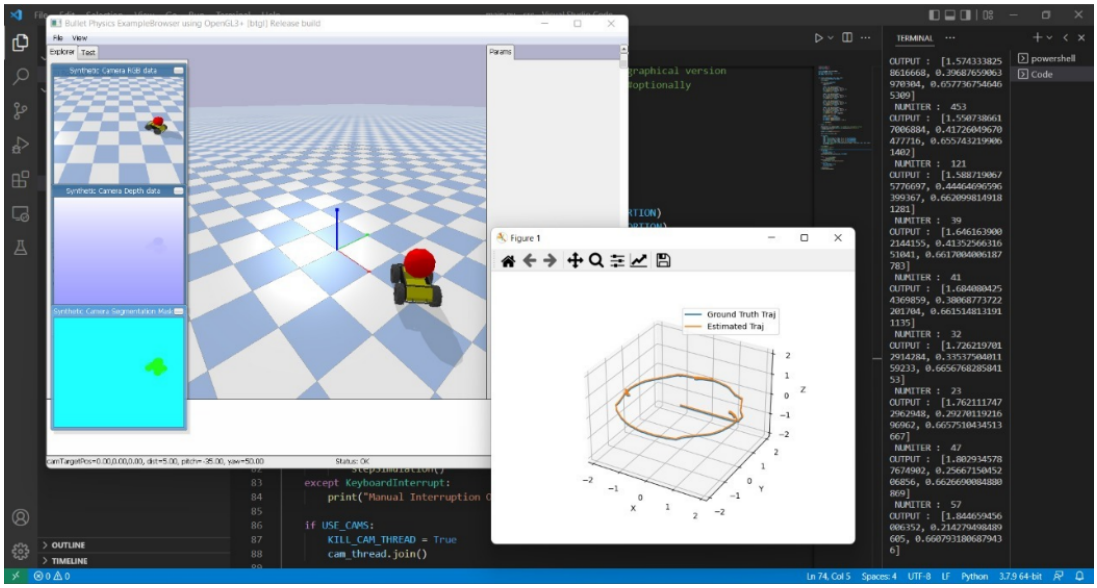

Figure 5: Simulation world and the plot of the trajectory of object.

5 References

1. https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html
2. https://in.mathworks.com/help/vision/ug/camera-calibration.html
3. https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/EPSRC_SSAZ/node3.html
4. https://www.sciencedirect.com/science/article/abs/pii/S1084804519302772
5. http://physbam.stanford.edu/cs448x/old/CameraParameters.html
6. https://www.sciencedirect.com/topics/engineering/pinhole-camera-model

7. https://www.pbr-book.org/3ed-2018/Geometry_and_Transformations/Transformations#TheLook-AtTransformation
8. https://www.songho.ca/math/homogeneous/homogeneous.html
9. https://builtin.com/data-science/gradient-descent