3D LOCALIZATION USING MULTI-CAMERA SYSTEM

A PREPRINT

Koneru Saketh
Department of Electrical Engineering
Indian Institute of Technology (BHU)
Varanasi-221005
koneru.saketh.eee19@itbhu.ac.in

Sagubandi Vishnu Murthy Naidu
Department of Electrical Engineering
Indian Institute of Technology (BHU)
Varanasi-221005
sagubandi.vmnaidu.eee19@itbhu.ac.in

Shaik Asif Ahmad
Department of Electrical Engineering
Indian Institute of Technology (BHU)
Varanasi-221005
shaikasif.ahmad.eee19@itbhu.ac.in

Vishwas Chepuri
Department of Electrical Engineering
Indian Institute of Technology (BHU)
Varanasi-221005
chepuri.vishwas.eee19@itbhu.ac.in

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

May 6, 2021

ABSTRACT

We consider three-dimensional (3D) localization of a mobile robot using a multi-camera sensor system. The set of calibrated and synchronized cameras are placed in fixed positions within the environment. We propose a real-time 3D localization algorithm which fuses the 2D image coordinates from multiple views synchronously in a multi-camera sensor system and determine the mobile robot's 3D location in an indoor environment. Moreover our solution works in real-time and is easily deployable. Experimental results show the proposed algorithm can achieve reliable, efficient, and real-time 3D localization in indoor environments.

***K*eywords** Three-Dimensional Localization · Indoor Environment · Multi-Camera System · Gradient Descent · Computer Vision · PyBullet

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.

Although the use of sensors within the environment requires the installation of an infrastructure of sensors and processing nodes, it presents several advantages, it allows reducing the complexity of the electronics onboard the robots and facilitates simultaneous navigation of multiple mobile robots within the same environment without increasing the complexity of the infrastructure. Moreover, the information obtained from the robots movement is more complete, thereby it is possible to obtain information about the position of all of the robots, facilitating cooperation between them.

The sensor system in this work is based on an array of calibrated and synchronized cameras. There are several methods to locate mobile robots using an external camera array. The proposal presented in this paper uses attached artificial landmarks to locate robots and the camera geometry to obtain the positions. It uses a set of calibrated cameras for which pinhole imaging principle applies, placed in fixed positions within the environment to obtain the position of the robots.

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 that the camera parameters are already known, so the parameters for α_{th} camera are focal length $f_\alpha = (f_\alpha^x, f_\alpha^y)$, distortion coefficients $K_1^\alpha, K_2^\alpha, K_3^\alpha, P_1^\alpha, P_2^\alpha$, position in real-world coordinates C_α.

The task can be broadly divided into five parts:
1. Locating the robot in each view.
2. Removing the effects of distortion.
3. Conversion from 2D camera frame(camera plane) to 3D camera frame.
4. Conversion from Camera frame to World frame.
5. Finding the optimal solution.

2.1 Locating the robot in 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. Now we apply contour detection [1] 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 shows the pipeline of this task.

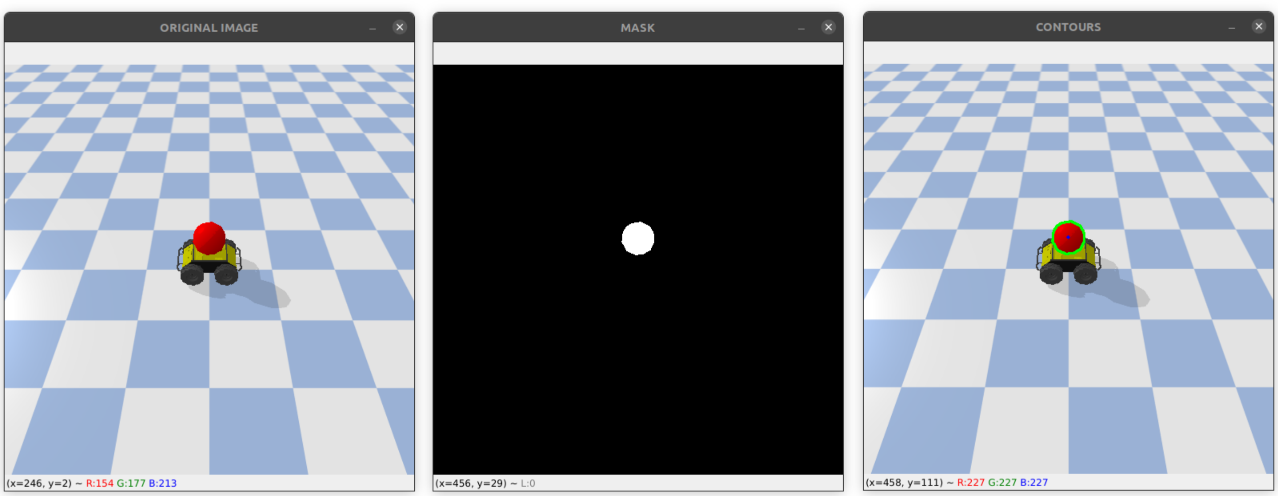

Figure 1: Masking and contour detection of a given image.

2.2 Removing the effects of distortion.

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

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

2.3 Conversion from 2D camera frame(camera plane) to 3D camera frame

For each view, we can project the 2D coordinates of the detected centroid from the image plane to a 3D point P_α in the camera frame in order to obtain a 3D line passing through the center of the camera and the point P_α and intersecting camera plane at the detected centroid. Figure 2 shows the projection from 2D camera coordinates to 3D camera coordinates.

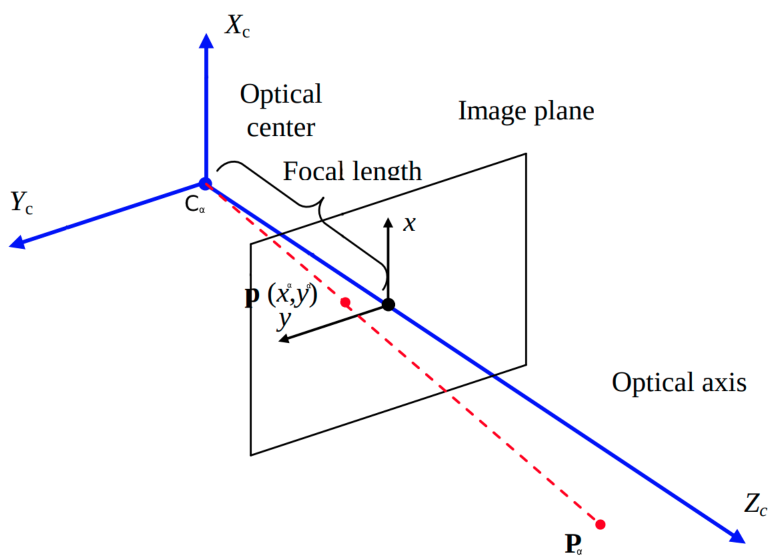

Figure 2: Conversion from 2D camera coordinates to 3D camera coordinates.

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

2.4 Conversion from Camera frame to 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 $\hat{\boldsymbol{U_\alpha}}$ 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 $\hat{\boldsymbol{D_\alpha}}$, $\hat{\boldsymbol{R_\alpha}}$ where $\hat{\boldsymbol{D_\alpha}} = L_\alpha - C_\alpha$ and $\hat{\boldsymbol{R_\alpha}} = \hat{\boldsymbol{U_\alpha}} \times \hat{\boldsymbol{D_\alpha}}$. We now generate transformation matrix as follows.

$$M_\alpha = \begin{bmatrix} \hat{\boldsymbol{R_\alpha}} & \hat{\boldsymbol{U_\alpha}} & \hat{\boldsymbol{D_\alpha}} & \mathbf{C}_\alpha \end{bmatrix} \tag{1}$$

where $\hat{\boldsymbol{R_\alpha}}, \hat{\boldsymbol{U_\alpha}}, \hat{\boldsymbol{D_\alpha}}, \mathbf{C}_\alpha$ are column vectors of length 3 and M_α is a 3×4 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 equation (2).

$$P_\alpha^R = M_\alpha \begin{bmatrix} P_\alpha^T \\ 0 \end{bmatrix} \tag{2}$$

2.5 Finding the 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.

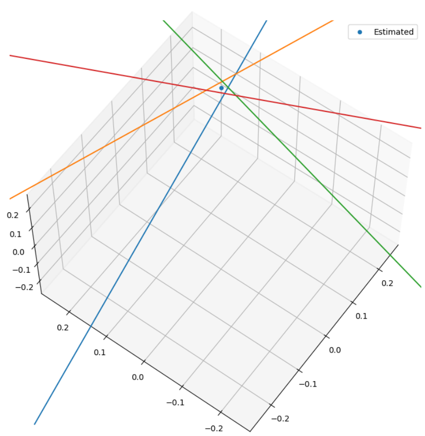

Figure 3: Four non-concurrent lines joining P_α^R and C_α for camera.

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.

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

3 Applications

3D localization using multi-camera systems can be used for surveillance of mobile objects in public spaces. It can also be used to monitor work environments. It can also be implemented in the field of entertainment and communication like digital animation. Other uses of this is in the fields of healthcare industry in the form of bio-mechanical analysis and clinical diagnosis.

4 Experiments and Results

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. Figure 4 shows our simulated world. We implemented the proposed approach in our simulated world to validate the results.

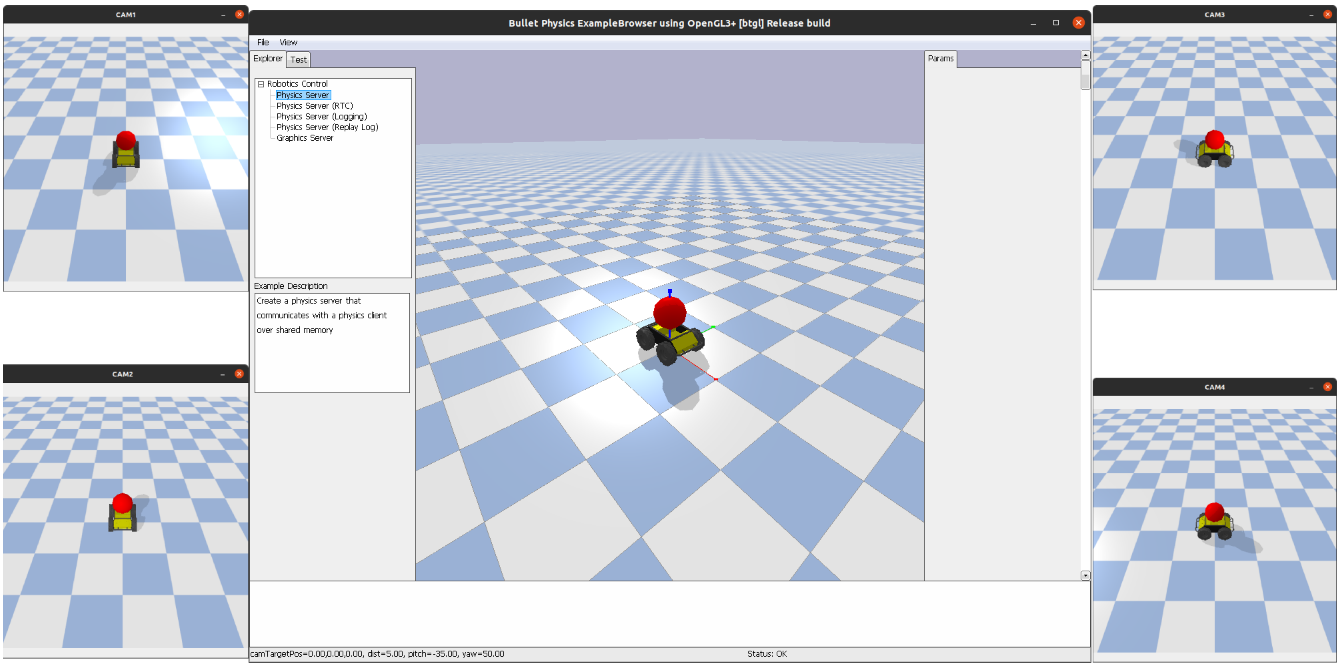

Figure 4: Simulated World

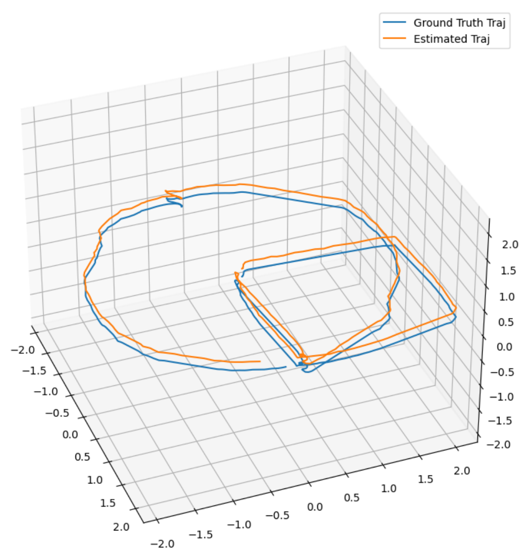

Figure 5: Plot shows ground truth and estimated trajectories of mobile robot.

References

[1] https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

[2] https://www.sciencedirect.com/science/article/abs/pii/S1084804519302772

[3] http://physbam.stanford.edu/cs448x/old/CameraParameters.html

[4] https://www.sciencedirect.com/topics/engineering/pinhole-camera-model

[5] https://staff.fnwi.uva.nl/r.vandenboomgaard/IPCV20162017/LectureNotes/CV/PinholeCamera/PinholeCamera.html

[6] https://in.mathworks.com/help/vision/ug/camera-calibration.html

[7] https://www.songho.ca/math/homogeneous/homogeneous.html

[8] https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/EPSRC_SSAZ/node3.html

[9] https://www.pbr-book.org/3ed-2018/Geometry_and_Transformations/Transformations#TheLook-AtTransformation