

Project Report: Localization of a Red Ball Using Stereovision and Control of an RRR Robot Arm in CoppeliaSim

Abstract

This project demonstrates the integration of computer vision and robotics in a simulated environment using CoppeliaSim. The primary objective is to calibrate virtual vision sensors, utilize stereovision to localize a red ball in the environment, and employ inverse kinematics to maneuver an RRR robot arm to reach the ball. The simulation environment provides a controlled setting to test and validate the algorithms for vision-based object localization and robotic manipulation.

1. Introduction

In robotics, the ability to perceive and interact with the environment is crucial for autonomous operation. This project focuses on two key aspects: object localization using stereovision and robotic arm control using inverse kinematics. The simulation environment of CoppeliaSim is utilized to model the scenario, allowing for the testing of algorithms without the need for physical hardware.

1.1 Objectives

- Calibrate virtual vision sensors to correct for lens distortions and ensure accurate image capture.
- Implement stereovision to determine the 3D position of a red ball in the environment.
- Develop and apply inverse kinematics to control an RRR robot arm to reach the localized ball.

2. Methodology

2.1 Simulation Setup

The project is implemented in CoppeliaSim, where the following components are modeled:

- RRR Robot Arm:** A three-degree-of-freedom (DOF) robot arm with revolute joints (RRR configuration).
- Virtual Vision Sensors:** Two cameras positioned to capture stereo images of the environment.
- Environment:** A simulated space containing a red ball as the target object.

2.2 Vision Sensor Calibration

Calibration of the virtual vision sensors is performed to correct for lens distortions and to compute the intrinsic and extrinsic parameters. The calibration process involves:

- Intrinsic Parameters:** Focal length, principal point, and distortion coefficients.
- Extrinsic Parameters:** Position and orientation of the cameras relative to the world coordinate system.

The calibration parameters are stored in a file (`stereoMap.xml`), which is loaded during runtime to undistort and rectify the captured images.

2.3 Stereovision-Based Object Localization

The stereovision technique is employed to estimate the 3D position of the red ball:

- Image Acquisition:** Capture synchronized images from the left and right cameras.
- Image Processing:**
 - HSV Filtering:** Apply HSV color filtering to segment the red ball from the background.
 - Blob Detection:** Identify the circular shape of the ball using contour detection.
- Disparity Calculation:**
 - Correspondence Matching:** Find the corresponding pixels of the ball in both images.
 - Depth Estimation:** Use the disparity and camera parameters to compute the depth (z-coordinate) of the ball.
- 3D Position Estimation:** Combine the pixel coordinates and depth to determine the 3D position of the ball in the world coordinate system.

2.4 Inverse Kinematics for Robot Arm Control

The inverse kinematics (IK) problem is solved to determine the joint angles required for the robot arm to reach the localized ball:

- World to Cylinder Coordinate Transformation:** Convert the ball's world coordinates to the coordinate system of the robot arm's base.
- IK Equations:** Derive the joint angles (θ_1 , θ_2 , θ_3) based on the arm's kinematic model and the target position.
- Joint Control:** Set the target positions for the joints to move the robot arm to the desired location.

3. Results

3.1 Vision Sensor Calibration

The calibration process successfully corrected for lens distortions, as evidenced by the rectified images. The intrinsic and extrinsic parameters were accurately determined, enabling precise stereovision calculations.

3.2 Object Localization

The stereovision algorithm effectively localized the red ball in the environment. The estimated 3D coordinates of the ball were:

```
Red_ball world coordinates: [x, y, z]
```

3.3 Robot Arm Control

The inverse kinematics solution successfully calculated the joint angles required to position the robot arm at the location of the ball. The robot arm moved to the target position with the following joint angles:

```
Theta 1  
Theta 2  
Theta 3
```

4. Discussion

4.1 Challenges

- **Learning CoppeliaSim and stereovision using opencv:** I had to watch many youtube videos to learn how to use CoppeliaSim and how to use openCV in the Calibration and stereovision tasks.
- **Inverse Kinematics Complexity:** Solving the IK equations for the RRR robot arm required careful consideration of the arm's kinematic constraints.

4.2 Potential Improvements

- **Enhanced Object Detection:** Implementing more robust object detection algorithms, such as machine learning-based methods, could improve localization accuracy.
- **Real-time Performance:** Optimizing the algorithms for real-time execution would be necessary for practical applications.

5. Conclusion

This project successfully demonstrated the integration of stereovision-based object localization and inverse kinematics-based robot arm control in a simulated environment. The results validate the effectiveness of the proposed approach in accurately localizing the red ball and maneuvering the robot arm to reach the target. Future work could focus on extending the approach to more complex environments and tasks.

6. References

- OpenCV Documentation. (n.d.). Retrieved from <https://docs.opencv.org/>
- CoppeliaSim User Manual. (n.d.).