Journal of Advanced Robotics, Autonomous Systems and Human-Machine Interaction

PRIME OPEN ACCESS

Volume 2, Issue 1

Research Article

Date of Submission: 23 Dec, 2025
Date of Acceptance: 12 Jan, 2026
Date of Publication: 22 Jan, 2026

Vision-Based Robotic Ball Tracking with Hough Transform and Modular 3D Estimation: A ROS2 Framework for Dynamic Environments

Oussama Errouji*

Euromed University of Fez, Morocco

***Corresponding Author:**
Oussama Errouji, Euromed University of Fez, Morocco.

Citation: Errouji, O. (2025). Vision-Based Robotic Ball Tracking with Hough Transform and Modular 3D Estimation: A ROS2 Framework for Dynamic Environments. *J Adv Robot Auton Syst Hum Mach Interact, 2*(1), 01-07.

Abstract

This paper introduces a ROS2-based robotic system for realtimeball detection, 3D localization, and autonomous tracking. Unlike traditional blob detection methods, the system employsthe Hough Circle Transform for robust 2D detection, coupled with a decoupled architecture where 3D estimation operates independently from motion control. The modular design enables simultaneous applications: reactive tracking via 2D data and 3D spatial analysis for external tasks. Evaluated in Gazebo simulations, the system demonstrates adaptability to varying lighting, ball sizes, and distances. Applications span automated ball collection, industrial sorting, and augmented reality interactions.

Keywords: ROS2, Hough Transform, Monocular Depth Estimation

Introduction

Robust detection and tracking of spherical objects in dynamic environments remain critical for robotics. This work advances
prior methods by:

- Leveraging the Hough Transform as a powerful and efficient algorithm for accurate circle detection, even under noisy conditions.
- Decoupling 3D estimation from tracking, enabling parallel processing and more flexible application scenarios.
- Validating system performance through systematic Gazebo benchmarks, including multi-ball tracking and varied lighting conditions. The system's ROS2-based architecture ensures scalability, while simulation-driven evaluation guarantees reproducibility and robustness.

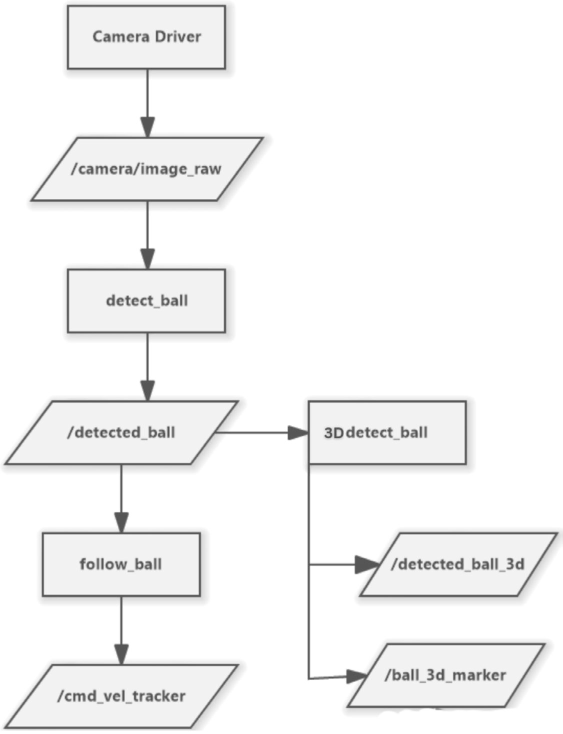

Figure 1: System Architecture

System Architecture

The pipeline comprises three independent nodes (Fig. 1):

- detect ball: 2D detection using Hough Transform (output:detected ball with u, v coordinates and apparentsize).
- detect ball 3d: 3D position estimation (output:detected ball 3d).
- follow ball: Reactive velocity control using 2D data(output: cmd vel).

Methodology

2D Detection Via Hough Transform

The detect ball node processes images from camera/image raw through the following pipeline:

- Preprocessing: A Gaussian blur is applied to reduce noise, followed by Adaptive Histogram Equalization to enhance local contrast and edge visibility.
- Color Masking (HSV): The image is converted to the HSV color space, and thresholding is applied to isolate the target hue range, making color-based segmentation more robust to lighting variations.
- Hough Circle Transform: Circles are detected using the gradient-based Hough Transform, which maps edge points into a three-dimensional accumulator space (a, b, r) corresponding to potential circle centers (a, b) and radii r. A circle with center (a, b) and radius r satisfies:

$$(x - a)^2 + (y - b)^2 = r^2$$

Preprint submitted to Elsevier May 27, 2025 Edge pixels vote in this parameter space, and peaks in the accumulator indicate likely circle candidates. Filtering is applied using:

- Minimum/maximum radius to discard irrelevant detections.
- Accumulator threshold to select high-confidence candidates.

- **Output:** The node publishes the normalized pixel coordinates (u, v) and the detected radius to /detected ball.

3D Position Estimation

The detect ball 3d node subscribes to /detected ball and calculates the 3D coordinates (x3d, y3d, z3d) using the following steps:

- **Depth Estimation from Apparent Size**

$$\theta_{\text{ball}} = z_{2d} \times \text{hfov} \tag{1}$$

$$d = \frac{r}{\tan\left(\frac{\theta_{\text{ball}}}{2}\right)} \tag{2}$$

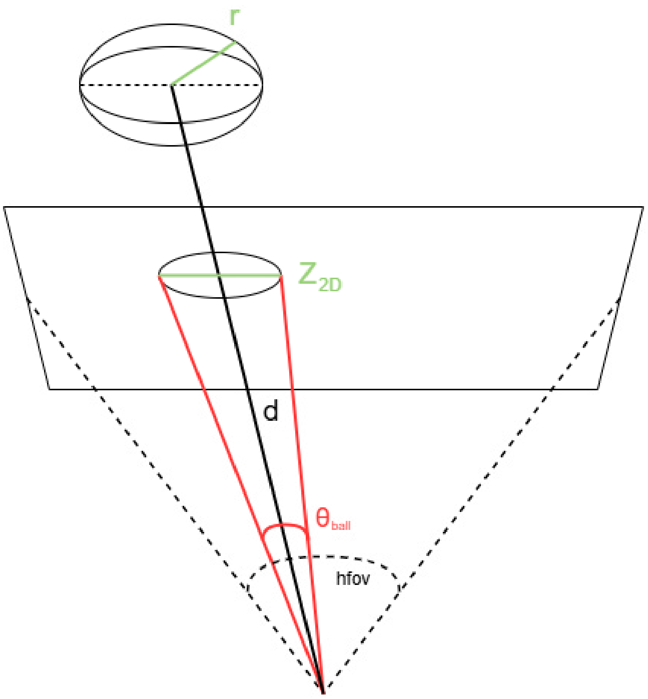

Figure 2: (a) Estimating Depth d from the Apparent Size and Horizontal Field of View

- **Vertical Angle and Y Coordinate**

$$\theta_y = y_{2d} \times \frac{\text{vfov}}{2} \quad (3)$$

$$y_{3d} = d \times \sin(\theta_y) \quad (4)$$

$$d' = d \times \cos(\theta_y) \quad (5)$$

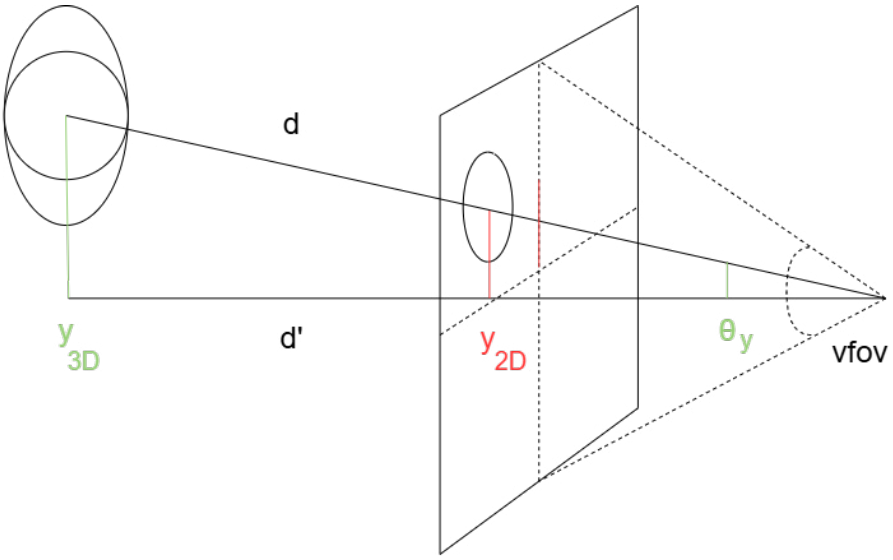

Figure 3: (b) Estimating Vertical Position y3d and Projected Distance d'

- Horizontal Angle and X/Z Coordinates

$$\theta_x = x_{2d} \times \frac{\text{hfov}}{2} \quad (6)$$

$$x_{3d} = d' \times \sin(\theta_x) \quad (7)$$

$$z_{3d} = d' \times \cos(\theta_x) \quad (8)$$

Reactive Tracking

The tracking system relies on a reactive closed-loop control approach that continuously adjusts the robot's motion based on real-time visual feedback of the detected spherical object. The methodology consists of two main control components:

- **Angular Control:** The system uses a proportional (P) controller to minimize the horizontal displacement between the ball's position and the center of the camera frame. Let $x \in [-1, 1]$ denote the normalized horizontal offset of the detected ball, where $x = 0$ indicates perfect alignment. The angular velocity ω is computed as:

$$\omega = -K_p \cdot x$$

where K_p is the angular gain (chase multiplier). The negative sign ensures corrective rotation in the opposite direction of the offset to realign the robot with the ball.

- **Linear Control:** The forward velocity v is controlled based on the perceived distance to the ball. Since no explicit depth information is available, the pixel radius r of the detected circle is used as an inverse proxy for distance.

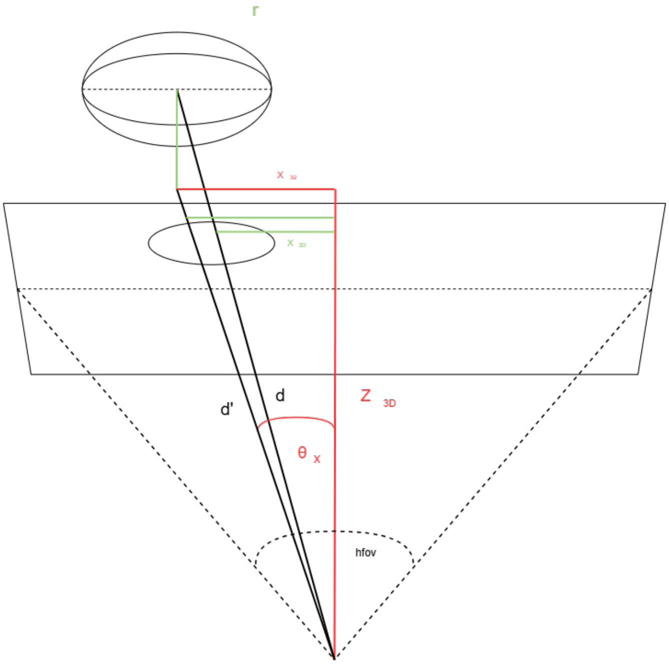

Figure 4: (c) Estimating x3d and z3d Using the Horizontal Angle

A threshold rmax is defined such that forward motion continues only when:

$r \ < r_{max}$

Under this condition, the robot moves forward at a constant velocity $v = v_f$, encouraging it to approach the object until it reaches the desired proximity.

- **Search Behavior:** In the absence of detection for a given timeout interval $\Delta t > t_{max}$, the robot enters a search mode. It spins in place with a fixed angular velocity ω_s to reacquire the target. This ensures robustness against temporary occlusions or detection failures. To smooth sudden changes and reduce noise in detections, exponential filtering is applied to both the horizontal offset x and the distance proxy r. At each time step t, the filtered estimate $\hat{x}_t$ is updated using:

$$\hat{x}_t = \alpha \cdot \hat{x}_{t-1} + (1 - \alpha) \cdot x_t$$

where α ∈ [0, 1] is the filter coefficient controlling memory. A similar equation is applied for the radius value r. This filtering mitigates the impact of detection jitter and enables more stable control commands. This approach offers a lightweight, real-time tracking solution suitable for embedded platforms and low-latency robotics applications.

Results and Analytics

This section presents the key experimental results and performance benchmarks obtained from the simulation environment. The evaluation focuses on robustness to lighting variations, spatial detection limits, control response, and trajectory tracking over time.

Detection Robustness Under Varying Illumination

To assess the system's reliability under different lighting conditions, we simulated varying light intensities by adjusting the diffuse RGB value of the scene's main light source. The corresponding approximate lux level was computed using:

$$\text{Lux} \approx \left(\frac{R+G+B}{3}\right) \times L_{\text{ref}} \tag{9}$$

Where:

- R,G, B ∈ [0, 1] are the normalized diffuse color components.
- Lref = 10,000 lux for standard white light.

The detection results are summarized in Table 1. The system reliably detects the ball even in low-light settings down to approximately 500 lux, demonstrating robustness to ambient lighting variations.

Diffuse RGB	Approx. Lux Level	Detected
`RGB(0.8, 0.8, 0.8)`	~8000 lx	True
`RGB(0.5, 0.5, 0.5)`	~5000 lx	True
`RGB(0.1, 0.1, 0.1)`	~1000 lx	True
`RGB(0.05, 0.05, 0.05)`	~500 lx	True

Table 1: Light Intensity (in lux) vs. Ball Detection Status

Spatial Detection Range

The perception module was tested to determine its operational range. The minimum and maximum distances at which the ball could be reliably detected were found to be:

- **Minimum distance:** 10 cm (due to field-of-view and distortion constraints).
- **Maximum distance:** 3.5 m (limited by camera resolution and circle size threshold).

This range is adequate for indoor robotic applications such as following, pick-and-place, or reactive interaction.

Detection Frame and Coordinate Output

Figure 5 shows a sample frame from the onboard camera with the detected ball and its 3D position output. The coordinates, expressed in the camera frame, are:

- x = 0.15m
- y = −0.34m
- z = 0.045m

This positional information is essential for downstream control and can be used for more advanced behaviors such as 3D pose estimation and manipulation.

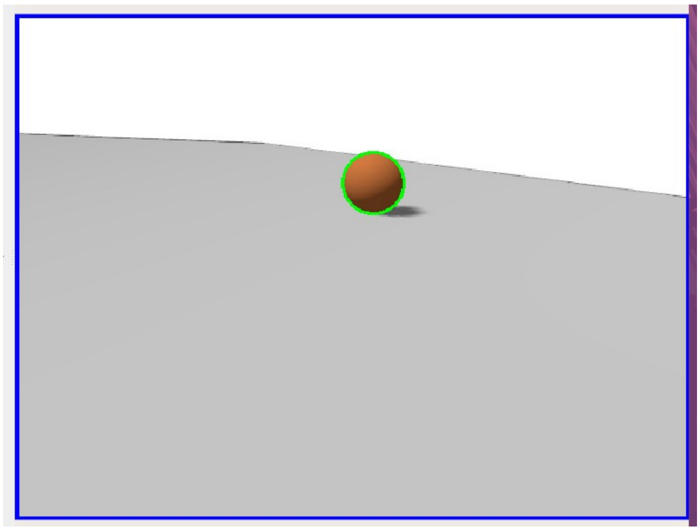

Figure 5: Ball Detected in Camera Frame with Overlayed Coordinates

Velocity Control Response

To evaluate the robot's motion behavior during tracking, we plotted the linear (v) and angular (ω) velocities as functions of time while approaching a detected ball. As seen in Figure 6, the robot starts rotating to align with the target and then moves forward steadily until reaching it.

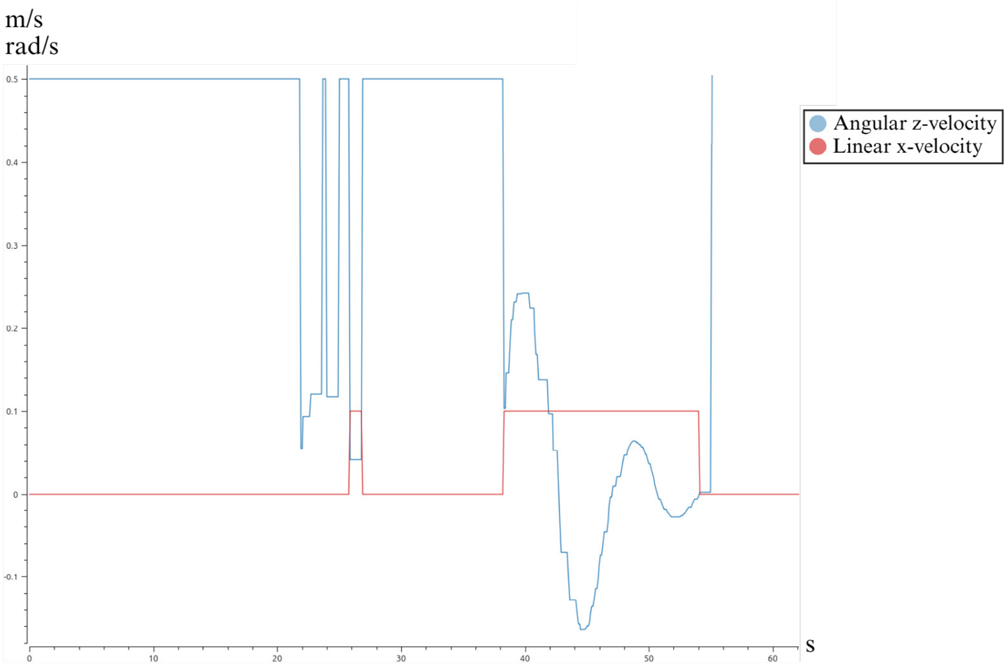

Figure 6: Linear (v) and Angular (ω) Velocities of the Robot During Ball Following

Corresponding simulation snapshots at initial time t0 and final time t1 illustrate the robot's successful approach and alignment with the ball, shown in Figure 7.

Trajectory and Coordinate Tracking Over Time

To analyze both detection stability and robot movement, we tracked the ball's coordinates (in the camera frame) and the robot's own position (in the world frame) over time. Figure 8 shows the temporal evolution of:

- The detected ball's x, y, z coordinates (camera-relative).
- The robot's global trajectory in x, y (world frame).

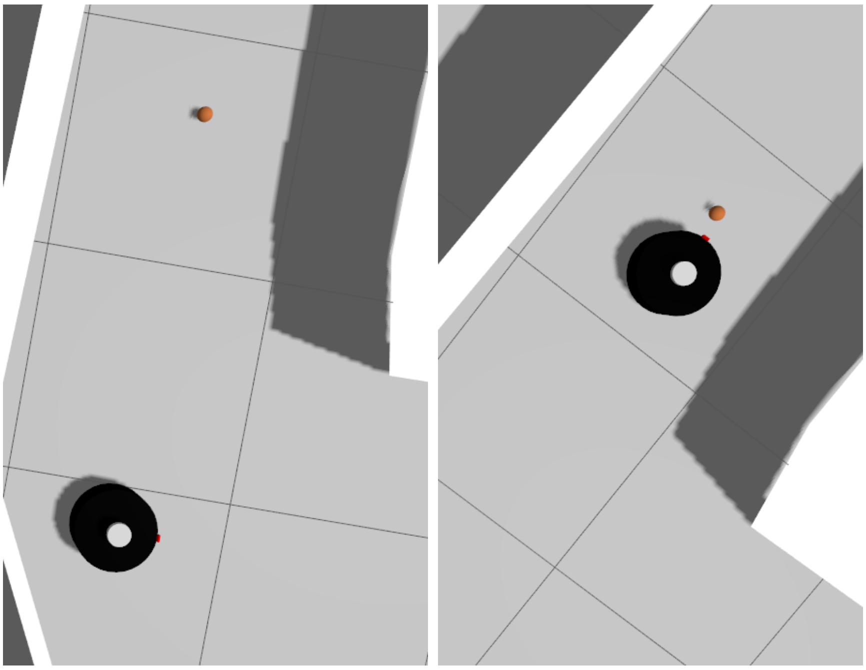

Figure 7: Robot at t0 (left) Before Tracking; at t1 (right) After Approaching the Ball

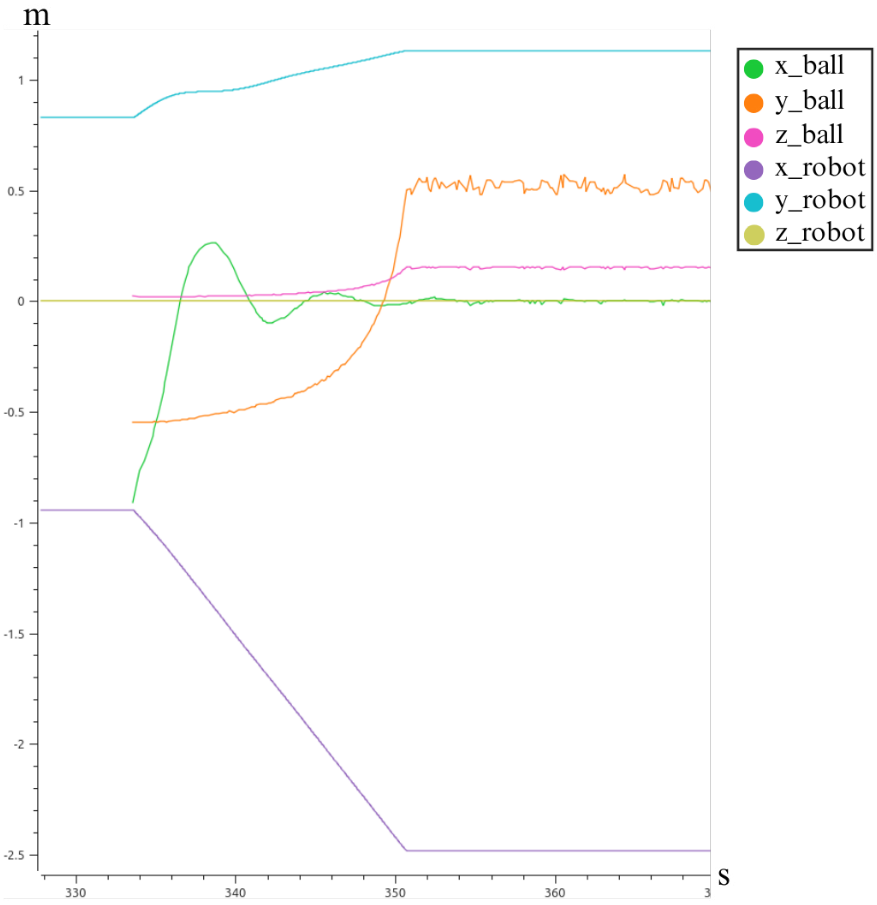

Figure 8: Time Evolution of Detected Ball Coordinates (Camera Frame) and Robot Trajectory (world frame)

The plot confirms smooth motion toward the ball, with consistent distance reduction and minimal drift in lateral position, validating the effectiveness of the control strategy. This completes the evaluation section, providing evidence of the system's robustness, control accuracy, and functional integration in realistic conditions.

Applications

- Automated Ball Collection: Robots gather sports balls in stadiums using 2D tracking.
- Industrial Sorting: 3D coordinates guide robotic arms to categorize objects by size/spatial position.
- Augmented Reality (AR): 3D data overlays virtual el Agricultural Robotics: Fruit-picking robots use sizefiltered Hough detection to harvest crops.

Conclusion

This work demonstrates a modular ROS2 framework for vision-based ball tracking, validated through rigorous simulation benchmarks. By decoupling 2D and 3D processing, the system supports diverse applications, from industrial automation to interactive AR [1].

References

1. Ungurean, I. (2020). Timing comparison of the real-time operating systems for small microcontrollers. Symmetry, 12(4), 592.