AI-Driven Single-Camera Vision System for Ground Object Localization

Jannatul Naim Apu

Department of Computer Science and Engineering

United International University (UIU)

Abstract

—This paper presents a real-time monocular vision system capable of detecting common objects and estimating their real-world ground-plane coordinates using a single RGB camera. The system integrates YOLOv8 for object detection with a geometric projection model that maps image pixels to physical coordinates under known camera constraints. Assuming objects rest on a flat ground plane, the system estimates X–Z positions in centimeters and exposes perception results via a RESTful API. Experimental evaluation shows that the system achieves practical accuracy for short-range robotic applications while maintaining real-time performance on low-cost hardware.

Index Terms—Monocular vision, object detection, YOLOv8, ground-plane projection, robotic perception.

I. Introduction

Robotic perception systems commonly rely on stereo cameras or depth sensors to estimate object positions in three-dimensional space. While effective, these approaches increase system cost, power consumption, and calibration complexity. Monocular vision offers a low-cost alternative but requires geometric reasoning to recover spatial information.

This work proposes a single-camera vision system that estimates ground-plane object coordinates using known camera parameters and real-time object detection. The system is designed for integration with robotic arms and autonomous agents where moderate accuracy is sufficient for grasping, navigation, and interaction tasks.

II. System Architecture

The overall system architecture is modular and multithreaded. It consists of five primary components:

1. **main.py** – System entry point and process coordinator
2. **config.py** – Centralized configuration parameters
3. **geometry.py** – Camera model and pixel-to-world transformation
4. **vision.py** – Camera capture and YOLO-based inference thread
5. **server.py** – Flask-based REST API

The vision pipeline executes in a dedicated thread, while the Flask server runs concurrently to provide non-blocking access to perception results.

III. Coordinate System and Assumptions

A right-handed ground-plane coordinate system is defined as follows:

- X-axis: Horizontal left/right displacement (cm)
- Z-axis: Forward distance from the camera (cm)
- Y-axis: Vertical axis, fixed at 0 assuming objects rest on the ground

The camera is mounted at a fixed height above the ground and tilted downward by a known angle. All detected objects are assumed to lie on a flat ground plane.

IV. Camera Geometry and Projection Model

A. Camera Intrinsics

The camera intrinsic parameters are derived from the image resolution and horizontal field of view (HFOV). The focal length is computed as:

F(x) = {W/2} / { tan (HFOV / 2) }

where (W) is the image width in pixels.

B. Pixel-to-World Mapping

For each detected object, the bottom-center pixel of the bounding box is assumed to correspond to the ground contact point. The vertical angle is computed relative to the optical axis and adjusted by the camera tilt angle. The forward distance (Z) is estimated using trigonometric projection:

Z = H / { tan (phi + theta) }

where (H) is the camera height, (\phi) is the pixel-derived vertical angle, and (\theta) is the camera tilt.

To improve stability, a secondary depth estimate is computed using the known real-world width of the object class. The final depth is obtained via weighted fusion of both estimates.

V. Object Detection

The system employs YOLOv8n from the Ultralytics framework due to its favorable trade-off between accuracy and speed. Inference is performed at fixed intervals to maintain stable frame rates. Only predefined target classes are considered to reduce false positives and computational overhead.

Each detection provides the object's class, confidence score, pixel location, and estimated real-world coordinates.

VI. Multithreading and API Design

A multithreaded design ensures real-time performance:

- The vision thread handles camera capture and inference
- The main thread hosts a Flask server
- Shared perception state is protected using a mutex lock

The REST API exposes a single endpoint, **/vision**, which returns the current perception state in JSON format. This design enables seamless integration with robotic controllers and higher-level reasoning modules.

VII. Experimental Results

A. Performance

The system was evaluated using a standard USB camera at a resolution of 640×480 pixels. The observed performance is summarized in Table I.

Table I

System Performance

Configuration	FPS
YOLOv8n (320 px)	18–25
With visualization	15–20
Flask enabled	Stable

B. Accuracy

Accuracy was evaluated by placing objects at known distances on the ground plane.

Table II

Distance Estimation Accuracy

Distance Range	Error
10–20 cm	±1–2 cm
20–40 cm	±2–4 cm

VIII. Discussion

The results demonstrate that monocular vision can provide sufficiently accurate spatial information for short-range robotic tasks. While the system cannot replace depth sensors for high-precision applications, it offers a low-cost and computationally efficient alternative.

IX. Limitations and Future Work

The current system assumes a flat ground plane and known object dimensions. Future work will focus on automatic camera calibration, adaptive object size estimation, temporal filtering, and integration with depth or stereo vision for enhanced robustness.

X. Conclusion

This paper presented a real-time single-camera vision system capable of detecting objects and estimating their ground-plane coordinates using geometric projection. The system achieves a balance between accuracy, speed, and simplicity, making it suitable for embedded robotic applications and academic research.

References

[1] J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” *Proc. IEEE CVPR*, 2016.

[2] Ultralytics, “YOLOv8 Documentation,” 2023.

[3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.