Group 64's Project Proposal: Monocular Object Distance Estimation Using YOLO Bounding Boxes

Thomas Stickland, Oliver (Chengze) Zhao, Larry (Yicheng) Liu
{sticklat,zhaoc59,liu854}@mcmaster.ca

1 Overview

Modern computer vision systems are remarkably good at recognizing and classifying objects in an image, thanks to deep-learning-based detectors such as YOLO (You Only Look Once). However, most object-detection networks operate purely in 2D image space: they can tell what and where something is in pixels, but not how far away it is in meters. In many real-world applications — autonomous robots, drones, surveillance cameras, driver-assistance systems, or even simple obstacle-detection prototypes — estimating the distance between the camera and detected objects is critical for decision-making.

Traditionally, accurate depth estimation requires stereo cameras, LiDAR, or structured-light sensors, which provide 3D geometry directly. But these sensors are costly, power-hungry, and sometimes impractical to deploy. Our goal is to explore what can be done using only a single, ordinary RGB camera — a monocular setup — combined with modern AI object detection.

2 Core Concept

This project aims to estimate the real-world distance of detected objects using a single RGB camera and the bounding boxes generated by a YOLO object detector. Instead of relying on stereo vision or depth sensors, the system leverages camera calibration and geometric relationships between image size, focal length, and object dimensions to infer depth information.

By combining computer vision geometry and deep-learning detection, the project demonstrates how an inexpensive, single-camera setup can perform practical distance estimation tasks suitable for robotics, surveillance, and autonomous navigation.

The core idea is to take YOLO's 2D detection output — which identifies object positions and sizes in image coordinates — and translate it into approximate metric distances through either the pinhole camera model (size-based estimation) or the ground-plane model (contact-point mapping).

3 Dataset

For this project, we will use the KITTI Depth Prediction dataset, part of the publicly available KITTI Vision Benchmark Suite (https://www.cvlibs.net/datasets/kitti/eval_depth_all.php). The dataset was collected by the Karlsruhe Institute of Technology and the Toyota Technological Institute using a stereo camera system and a Velodyne HDL-64E LiDAR sensor mounted on a moving vehicle. Each sample in the dataset contains a RGB image captured by the cameras and a corresponding dense depth map generated from LiDAR measurements. In addition to the depth maps, we will use KITTI's 2D object annotations (bounding-box coordinates x_1, y_1, x_2, y_2) to define object instances per image; each instance is paired with a scalar distance label derived from the dataset's depth information. The RGB image serves as the input (X) for our model, while the depth map acts as the label (Y), representing the ground-truth distance between each pixel in the image and the camera. These depth maps are provided as part of the dataset and were automatically generated by projecting LiDAR 3D points into the 2D camera image plane, eliminating the need for manual labeling. Operationally, for each annotated bounding box we treat the provided per-object depth (when available) as the target distance z; if an object's depth is not directly provided or is marked invalid, we fallback to the median of valid depth pixels within the box in the corresponding depth map.

Since the KITTI dataset is open-source and publicly available, no scraping or API access is required, and all data usage complies with KITTI's academic research license and terms of service.

The dataset also includes metadata such as sensor calibration parameters, camera poses, timestamps, and environmental context, which enhance reproducibility and model alignment. As the full dataset consists of over 93,000 stereo image pairs, we plan to use a 10–20% subset to reduce computational load while maintaining coverage across diverse environments (urban, rural, and highway). The dataset's high-quality annotations and real-world depth data make it ideal for supervised training of our vision-based obstacle detection and distance estimation model for autonomous drones. Unlike datasets without labels—which require manual annotation of objects or distances—KITTI provides pre-labeled, LiDAR-verified ground-truth depth values, ensuring accuracy and efficiency in the training process.

Example data records

Each row corresponds to one KITTI bounding box paired with a scalar distance target z (meters).

filename	**class**	**bbox** (x_1, y_1, x_2, y_2)	z **(m)**
003871.txt	Car	(276.07, 184.25, 342.82, 222.25)	30.69
003033.txt	Car	(724.61, 169.24, 794.58, 195.29)	43.21
006783.txt	Van	(606.32, 168.46, 622.08, 187.14)	93.99

4 Proposed Solution

We plan to solve this problem by using a combination of object detection and regression models. We will first begin with pre-labelled images to remove the object detection work. Once we have proven that our model works with this pre-labelled data, we can add object detection into our pipeline. For this, we would use an object detection model like YOLO11 to identify objects in the image and their bounding boxes, then use our model to estimate the distance to each object based on the bounding box coordinates and the object's class(Franke et al., 2021). If successful, this will allow us to estimate the distance to multiple objects in an image.

To build upon this first step and further improve the accuracy of our model, we could add orientation data to our feature set. This would allow us to account for the 3D orientation of objects in the image, which would help us to better estimate the distance to objects that are not facing directly towards the camera. We would follow the same process as before: first using pre-labelled data to prove the concept, then adding in a model to identify objects and their 3D orientations.

We plan to first try a Multi-Layer Neural Network for our regression model, as similar projects such as Disnet have found success with this approach(Haseeb et al., 2018). This could be expanded to be part of the YOLO CNN itself, by adding additional output layers to predict distance and orientation in addition to the bounding box coordinates and class(Kiefer et al., 2023).

Some libraries we plan to use include:

- PyTorch or Tensorflow for building and training our models (we will decide which one to use after an initial evaluation of both)
- Pretrained YOLO11 models for object detection & classification (ultralytics library)
- Numpy and Pandas for data manipulation
- Matplotlib and seaborn for data visualization

References

Marten Franke, Vaishnavi Gopinath, Chaitra Reddy, Danijela Ristic-Durrant, and Kai Michels. 2021. Bounding box dataset augmentation for long-range object distance estimation. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, page 1669–1677.

Muhammad Abdul Haseeb, Jianjun Guan, Danijela Ristić-Durrant, and Axel Gräser. 2018. Disnet: A novel method for distance estimation from monocular camera.

Benjamin Kiefer, Yitong Quan, and Andreas Zell. 2023. Approximate supervised object distance estimation on unmanned surface vehicles. *arXiv preprint arXiv:2501.05567v1*.