Fusion of information from 2D Lidar and RGB Camera

Prathmesh Gaonkar B20ME032
Mechanical Engineering
Indian Institute Of Technology Jodhpur
Jodhpur, India
gaonkar.1@iij.ac.in

Satyam Kumar Gupta B20ME068
Mechanical Engineering
Indian Institute Of Technology Jodhpur
Jodhpur, India
gupta.81@iij.ac.in

***Abstract*—The automotive industry is currently undergoing a transformative phase with the rapid adoption of autonomous systems. Object detection plays a pivotal role in ensuring safe navigation, and Convolutional Neural Networks (CNNs) have emerged as a cornerstone technology in this domain. Our project is focused on fusion of 2D LiDAR and RGB image data explicitly tailored for object detection in autonomous vehicles. We have tried implementing the YOLOv5 algorithm on the KITTI dataset which has 2D LiDAR point cloud and RGB data of various road scenes to achieve this task. This model is designed to excel in accurately identifying and localizing a diverse range of objects in real-time scenarios. Its capabilities extend beyond merely detecting cars to encompass pedestrians, cyclists, obstacles, and other relevant entities crucial for safe navigation. By harnessing the power of deep learning, we aim to push the boundaries of object detection in autonomous driving.**

***Index Terms*—Lidar, RGB Camera, Robotics**

I. Introduction

Combining 2D LiDAR and RGB camera data has become a powerful method for improving environmental awareness in mobile robotics. Although 2D LiDARs have historically provided depth data, their maps are frequently sparse. On the other hand, RGB photographs lack direct depth information but offer rich visual context. By utilizing the precise depth measurements provided by LiDARs and the contextual richness of RGB photos, this study aims to connect two modalities and improve the completeness of sparse LiDAR maps. The technique seeks to increase depth prediction accuracy by merging many sources, which is important for the autonomy of robotics and vehicles.

A revolutionary technique called BEVFusion is introduced in one of the works done. [1] This innovative LiDAR-Camera fusion framework is designed for 3D object detection in autonomous driving scenarios. It can operate independently even when LiDAR input is not available. It outperforms current approaches, performs better under typical circumstances, and significantly increases resilience. It can handle a wide range of LiDAR failures without the need for extra post-processing. The architecture consists of two separate streams that separately encode camera and LiDAR inputs into BEV space characteristics, then combine them to improve perception precision. BEVFusion exhibits promising potential for real-world implementation, with a strong focus on robustness against LiDAR failures and generalization across various architectures.

In another paper it proposes a novel technique for 3D object detection and localization using only a single image. [2] Unlike existing methods focusing solely on predicting object orientation, this approach combines deep CNN predictions with geometric constraints from 2D bounding boxes to generate precise 3D bounding boxes. The method utilizes a hybrid discrete-continuous loss function for orientation estimation, surpassing L2 loss performance significantly. By incorporating predictions and geometric constraints, it achieves reliable and accurate 3D object positioning. Evaluation of the KITTI benchmark demonstrates its effectiveness in 3D orientation estimation and bounding box correctness

II. Theory

A. LiDAR Sensor

LIDAR sensors work by emitting laser pulses and measuring the time it takes for the pulses to return after hitting objects in the environment. By analyzing the timing and direction of these pulses, LiDAR sensors can generate detailed 3D point clouds, where each point represents a reflection from a surface in the sensor's field of view. These point clouds provide rich spatial information about the surrounding environment, including the positions and shapes of objects. In the data collection for KITTI dataset Velodyne lidar sensors are used[3]. Velodyne LiDAR sensors are commonly used in various applications such as autonomous vehicles, robotics, and 3D mapping.

B. YOLO Algorithm

In our project, we have tried using the YOLO algorithm for object detection. Well, when it comes to object detection, at this time, we don't only have to identify the object, but we need to locate the object as well. At the same time, we need to figure out the object's size too; that's why we have b_x, b_y that describes the centroid of the bounding box and b_h, b_w describe the height and width of the bounding box. So, suppose we want to check whether the given image has a pedestrian, motorcycle, or car, the first thing is to check is there any of these cases, so if the image has any one of these, we label it as p = 1, otherwise if it does not have any of the above p = 0, now the next step is to check if p = 1, then among

pedestrian, motorcycle, and car, which of these is present for that, we have labeled as c_1, c_2, c_3 and if any of these are resent the corresponding label will be assigned 1 and rest are zero. Now, after the identification of the type of object, we need to calculate the dimension of the bounding box and its centroid, so the label will turn out to be: Y = [p_c, b_x, b_y, b_h, b_w, c].

Now comes the part of how well the object localization is actually happening, to evaluate that we use the Intersection over Union (IoU) method the idea is that we take the ratio of the areas of intersection of the ground truth bounding box to the predicted bounding box to the area of union of ground truth and predicted bounding box, as shown in the figure below. If IoU is greater than 0.5, the object prediction is good otherwise is poor. Now, it also happens that the algorithm may detect the same object multiple times, so to handle such a problem, there is a technique called Non-max suppression. Basically, you can see in the below image that the same object has been detected multiple times, now every object detection is associated with an IoU, so whichever IoU is less than the largest IoU that bounding box is suppressed, as you can see the bounding box with the highest IoU is highlighted most and rest will be discarded.

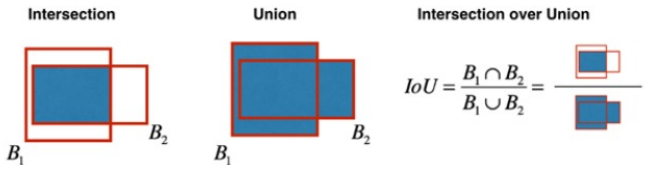

Fig. 1. Intersection over Union

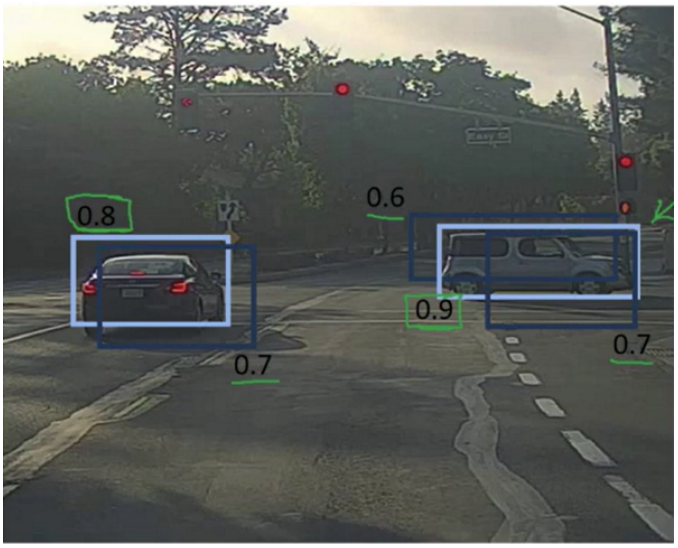

Fig. 2. Non-max supression

III. Method

A. Data preprocessing and LiDAR integration

Our proposed method demonstrates a comprehensive pipeline for processing LiDAR point cloud data and integrating it with object detections from images to draw 3D bounding boxes around detected objects. The process begins with downloading and preparing the KITTI dataset, focusing on a specific dataset containing synchronized raw city data. Key components include retrieving object detections from images, obtaining LiDAR point clouds, and removing the ground plane via RANSAC. The LiDAR points are then clustered in LiDAR space and associated with detected objects in image space. Camera calibration data is extracted to obtain projection matrices and rectified rotation matrices, enabling the transformation of LiDAR points to camera frame of reference. The LiDAR to camera rotation and translation matrices are also computed for further transformation. The pipeline includes functions to convert LiDAR points to camera space, project LiDAR point clouds onto the image coordinate frame, and obtain depth measurements for each detected object.

B. Object detection using YOLOv5 and 2D LiDAR

Object detection is performed using YOLOv5, with confidence and IOU thresholds set for optimal detection. Once detections are obtained, distances to objects are calculated and drawn on the image. LiDAR points are then clustered using DBSCAN, and clusters are transformed into image space. Further processing involves refining clusters, drawing 3D bounding boxes around clusters, and associating them with detected objects.
The pipeline is tested on sample images and LiDAR data from the KITTI dataset, demonstrating its effectiveness in drawing 3D bounding boxes around detected objects. Finally, the pipeline is extended to create a video by processing a sequence of images and LiDAR data. The resulting video showcases the integration of LiDAR point clouds with object detection to generate a visual representation of 3D bounding boxes around detected objects in a dynamic environment.

IV. Results

Firstly we have loaded the RGB information and Lidar sensor data in the same frame. The resulting figure is fig 3.

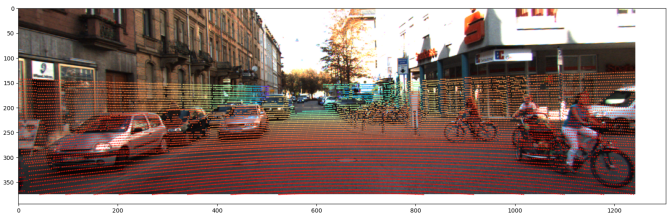

Fig. 3. RGB+Lidar information obtained from sensors in the same frame

We apply the RANSAC algorithm which discards the Lidar data corresponding to ground plane where no object is to be detected. As shown in Fig 4 the lidar data containing ground

points is vanished. This results into reduction of noise and computational complexity.

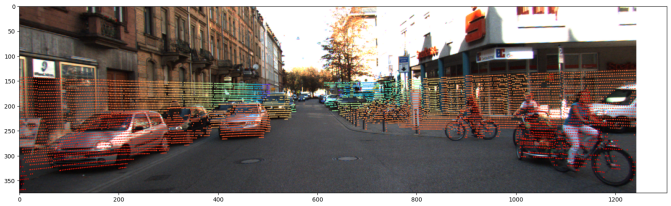

Fig. 4. Output after applying RANSAC on Fig

The image is fed into pre trained YOLOv5 model and resulting output results into bounding boxes about the detected objects as shown in fig 5.

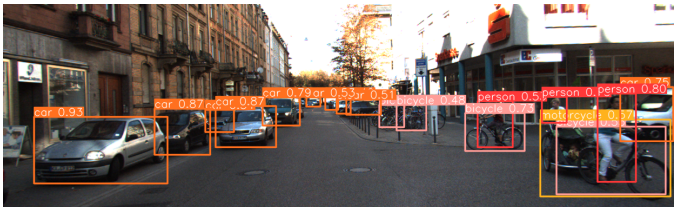

Fig. 5. Object detection using YOLO

The distance of the detected object from the sensor is obtained by finding the closest lidar point near the center of the box and assigning the corresponding distance of the lidar point to the object. The resulting output is shown in Fig 6.

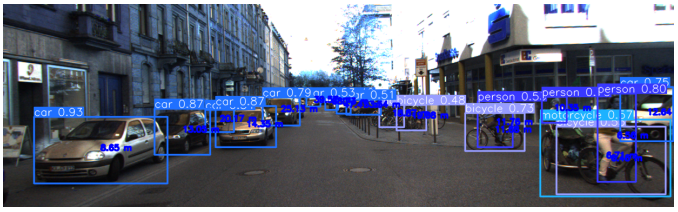

Fig. 6. Distance of the object from sensor using Lidar data

In the next part we are trying to obtain optimum object detection using fusion of Lidar data and RGB Camera. This is done using clustering algorithm on the lidar data points. We have used DBSCAN algorithm to cluster the lidar data. The resulting output is shown in Fig 7.

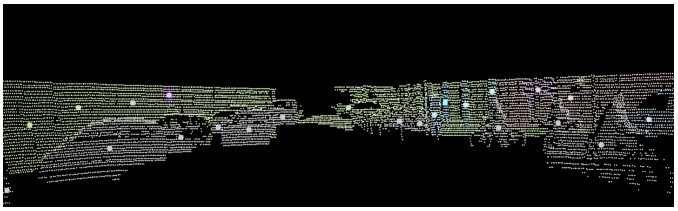

Fig. 7. DBSCAN Clustering on lidar data

Next we use the RGB data and lidar clustering to obtain optimum object detection. For each LiDAR cluster, determine its likelihood of containing an object based on its proximity to detected objects obtained using YOLO bouding boxes. This can be done by calculating the distance between the centroid of each LiDAR cluster and the centroids of detected objects in the camera image. If the distance is below a certain threshold, consider the LiDAR cluster as a likely candidate for an object. Apply additional criteria, such as cluster size or shape, to further filter and refine the likely clusters. Final clustering result obtained is shown in Fig 8.

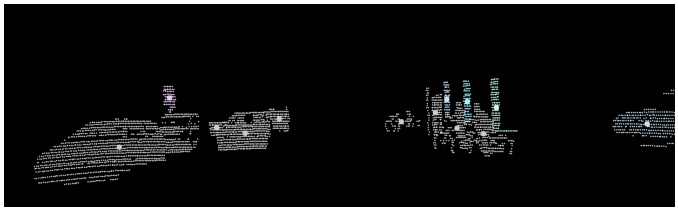

Fig. 8. Updated clustering on image

Finally based on the updated clustering, 3D bouding boxes are generated on the image as shown in Fig. 9.

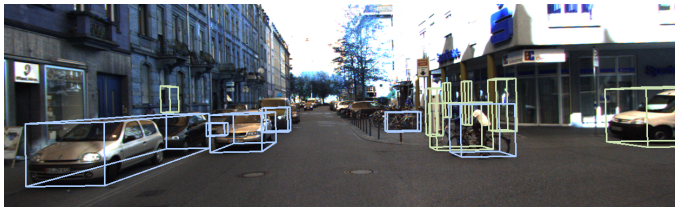

Fig. 9. Updated bounding boxes on image

By removing small, large, and distant clusters, the function reduces the amount of data that needs to be processed in downstream tasks such as object detection or tracking. This can lead to efficiency improvements in algorithm runtime and memory usage. Also, by focusing on likely clusters that are close to detected objects, the function improves the accuracy of subsequent processing steps. It reduces the likelihood of false positives and ensures that attention is directed towards clusters that are more likely to represent actual objects in the scene.

Video detecting objects using Lidar and RGB data - Click here
Link to code - Click here

V. Conclusion

Integrating LiDAR data with RGB camera data for object detection offers several advantages that can significantly enhance detection performance. LiDAR provides accurate depth information, which complements the RGB camera's 2D visual data. LiDAR data also helps filter out false positives by validating detected objects based on their 3D geometry and spatial context. By integrating information from both LiDAR and RGB camera sensors and leveraging clustering techniques to identify likely object candidates, this approach aims to

achieve optimum object detection performance, combining the strengths of both sensor modalities for enhanced perception capabilities.

VI. Contributions

Prathmesh Gaonkar : Data Preprocessing and LiDAR integration.
Satyam Kumar Gupta : Object detection.

References

[1] Ruibin, Z., Guo, Y., Long, Y., Zhou, Y., Jiang, C. (2022). Vehicle motion state prediction method integrating point cloud time series multiview features and multitarget interactive information. Journal of Advanced Transportation, 2022, 1-21. https://doi.org/10.1155/2022/4736623

[2] Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J. (2016). 3D Bounding Box Estimation Using Deep Learning and Geometry. ArXiv. /abs/1612.00496.

[3] https://www.cvlibs.net/datasets/kitti/

[4] Girshick, R. (2015). Fast R-CNN. ArXiv. /abs/1504.08083.

[5] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. ArXiv. /abs/1506.02640.