Image segmentation, Tracking, and Reconstruction of Dynamic Objects using Stereo Camera

Mohammed Maaruf Vazifdar

Master of Engineering, Robotics
University of Maryland
UID:117509717, maaruf98@umd.edu

Ashutosh Reddy Atimyala

Master of Engineering, Robotics
University of Maryland
UID:118442129, atimyala@umd.edu

Maitreya Kulkarni

Master of Engineering, Robotics
University of Maryland
UID:117506075, mkulk98@umd.edu

Abstract

The aim of the project is 3D object detection and pose estimation using stereo images. The idea is to gain knowledge about the conventional methodology in segmenting the object from the given scene, estimating the pose of the object from its point cloud data and providing the 3D bounding box for object orientation and alignment with respect to the camera which provides the accurate 3D pose of the object. This described methodology is evaluated on KITTI dataset- Stereo Evaluation 2015.

1 Introduction

In recent years with the development of technologies in computer vision and deep learning, numerous impressive methods are proposed for accurate 2D object detection. However, beyond getting 2D bounding boxes or pixel masks 3D object detection is eagerly in demand in many robotic and autonomous driving applications as this method describes the object in more realistic way. Typical image-based segmentation 3D object detection approaches adopt the pipeline are like 2D detectors and mainly focused on RGB features extracted from 2D images, however, these features are not suitable for 3D related tasks because of loss of spatial information. With the rise of automated driving large-scale 3D reconstruction and semantic mapping are widely used. Most of these approaches are developed for Lidar and depth cameras which are not particularly suitable for large-scale outdoor environments. A 3D surface can be observed by using multiple camera images from different views and poses, an optimal patch selection algorithm can be implemented for optimal semantic class segmentation. A semantic segmentation model is used to differentiate vehicles from the scene, then a point cloud model is generated based on disparity map between right and left frames to get the estimation of distance from vehicles.

An intuitive solution is to use a CNN to predict the depth maps and then use these as input if we do not have any depth data available. The benefits of transform depth map into point cloud data can be enumerated as follows: (1) Point cloud data shows the spatial information explicitly, which make it easier for network to learn the non-linear memory mapping from input to output. (2) Richer features can be learnt by the network because some specific spatial features exist only in 3D space. (3) The recent significant progress of deep learning on point clouds provides a solid building brick, which we can estimate 3D detection results in a more effective and efficient way.

2 Related Work

We briefly review existing works on 3D object detection tasks based on Lidar and images in autonomous driving scenario. **Image-based 3D Object Detection:** In the early works, monocular-based methods share similar framework with 2D detection [1], but it is much more complicated for estimating the 3D coordinates (x, y, z) of object center, since only image appearance cannot decide the absolute physical location. Mono3D [3] and 3DOP [2] focus on 3D object proposals generation using prior knowledge (e.g., object size, ground plane) from monocular and stereo images, espectively. Deep3DBox [4] introduces geometric constraints because the 3D bounding box should fit tightly into 2D detection bounding box. Deep MANTA [1] encodes 3D vehicle information using key points, since they are rigid objects with well-known geometry. Then the vehicle recognition in Deep MANTA can be considered as extra key points detection. There are some methods that propose some effective prior knowledge or reasonable constraints, they fail to get better performance because of the lack of spatial information. Another recently proposed method [5] for monocular 3D object 3D object detection introduces a multi-level fusion-based scheme utilizes a stand alone module to estimate the disparity information and fuse it with RGB information input data encoding, 2D box estimation and 3D box estimation phase, respectively.
Lidar-based 3D Object Detection: Our implementation is on stereo image data; we transform the data representation into point cloud which is same to Lidar-based methods. So, we also introduce some typical approach based on Lidar. MV3D [5] encode 3D point clouds with multi-view feature maps, enabling region-based representation for multimodal fusion. With the development of deep learning on raw point clouds.

3 Proposed Method

In this section, we describe the proposed framework for stereo-camera based 3D object detection.

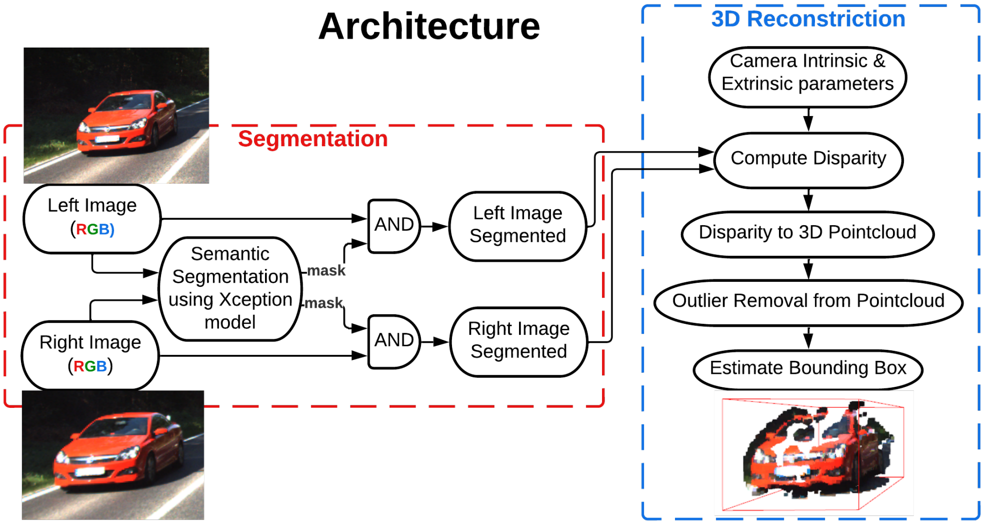


Figure 1: Project Architecture

3.1 Data-set Selection:

The dataset selected to evaluate the project pipeline is Stereo Evaluation 2015 dataset from KITTI dataset which has 200 training scenes and 200 test scenes. Based on the requirement for the proposed scenario, we select a few scenes from the combined training and test scenes with a removal of scenes with no objects to segment. These selected scenes were rectified before the use in segmentation model.

3.2 Semantic Segmentation using Xception model from DeeplabV3 framework:

Semantic segmentation is used to label each pixel of the image with a corresponding class. For segmenting the object from the selected scene, Xception model from DeeplabV3 framework trained on pascalvoc dataset is used. Xception is a CNN with 71 deep layers. The pre-trained model is trained on millions of images on pascal voc dataset with 20 object classes such as car, aeroplane, cat, person, house, motorbike, sheep etc with their respective colormaps.

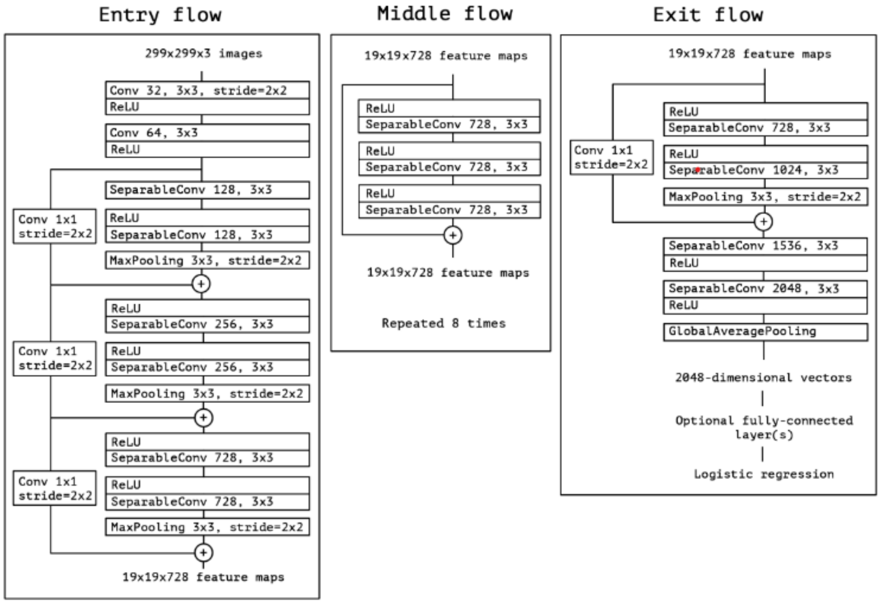


Figure 2: Architecture of Semantic Segmentation

This model uses the Depth Separable Convolutions concept consisting of two process, Depth wise Convolution: For an input image of size (weight x height x depth), a dxdxc kernel is used for convolution where d is the size of the kernel. Here, the convolution is performed only on 1 by 1 channel which results in a convolution of size kxkxc. Pointwise Convolution: A 1x1 kernel is used which iterates through every point. The depth of the kernel is the same as the number of channels of the input image.

The above architecture shows that the data is entered through the entry flow layer which has 36 convolutional layers for feature extraction. The data is then passed through the middle layer with separable convolutional layers which is repeated 8 times. As the goal is to classify images, the first layer is a convolutional base, followed by a logistic regression in the last layer (Exit flow). For the project, we use this Xception model for semantic segmentation to segment the car in the given input image. The segmented car mask from both the left and right stereo images is overlaid on the input image to get the segmented car image.

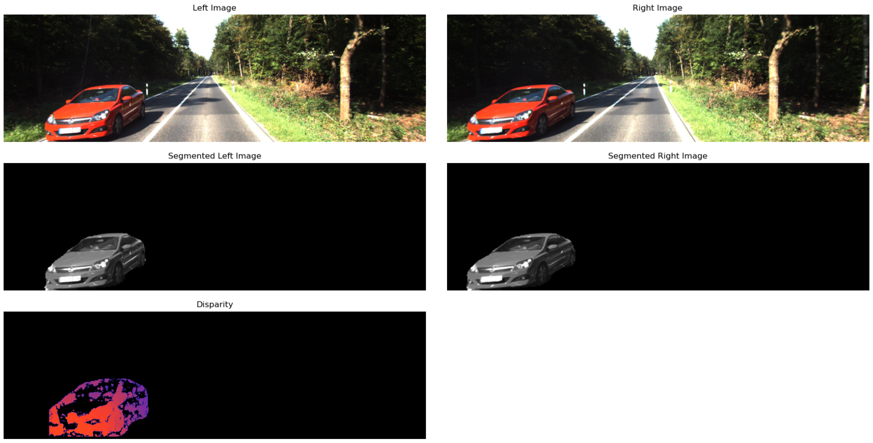


Figure 3: Segmentation Outputs and Disparity Map

3.3 Disparity, Depth Estimation and 3D PointCloud Construction

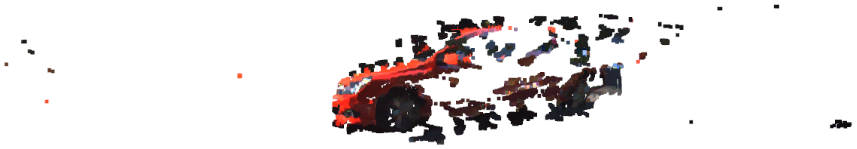


Figure 4: Initial PointCloud with outliers

The disparity map is estimated by StereoBM() in OpenCV using the left and right segmented grayscale images. By giving camera intrinsic and extrinsic parameters to OpenCV's stereoRectify, the disparity-to-depth matrix is calculated (). The 3D points are obtained using reprojectImageTo3D() and finally, colors are extracted from the left image and overlaid on the PointCloud. Once the outliers in the point cloud data are filtered, we put bounding boxes over the vehicles. Using the axis aligned boundingbox function from the open3d geometry library, a the bounding box is displayed over the PoinCloud.

3.4 Outlier Removal using Radius

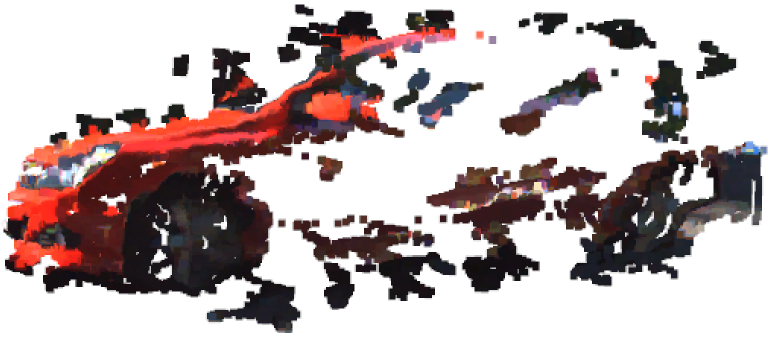


Figure 5: PointCloud with Outlier Removal

To properly determine the object's pose, the outliers in the PointCloud must be removed. Using the Open3d library's radius outlier removal function, a minimum number of nearby points that each point must have inside a specified sphere's radius in order to be kept in the PointCloud is used to obtain the conditioned PointCloud.

3.5 Bounding Box

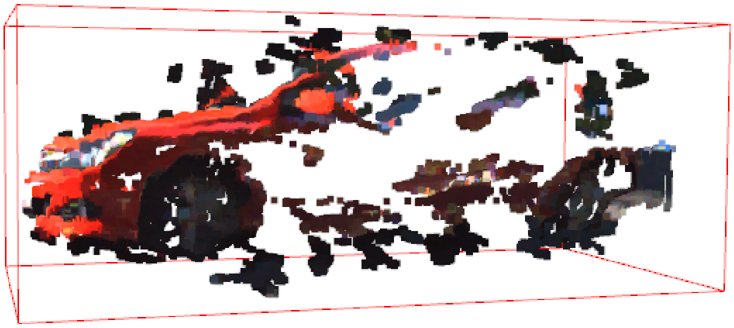
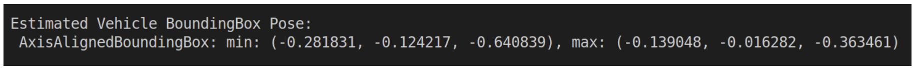


Figure 6: 3D Reconstruction of Vehicle



```
Estimated Vehicle BoundingBox Pose:
 AxisAlignedBoundingBox: min: (-0.281831, -0.124217, -0.640839), max: (-0.139048, -0.016282, -0.363461)
```

Figure 7: Vehicle Boundingbox Pose w.r.t camera

Bounding Box denotes the coordinates of the border of the cuboid enclosing the object segmented. This box can be further used to identify the pose for object detection. Using the get axis aligned bounding box function from the open3d geometry library, the bounding box is displayed over the PointCloud. The bounding box is generated using the PointCloud data with the inliers, which takes the minimum and maximum coordinates of the points which are defined as x,y,z coordinates, a 3D bounding box is generated enclosing the object.

4 Results

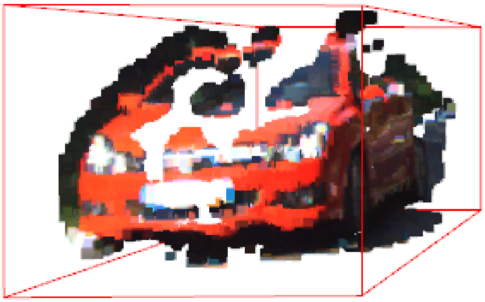


Figure 8: Result

Using the stereo images from the KITTI dataset and following the project architecture we have obtained the 3D reconstruction of the vehicle. The boundingbox and its pose is also estimated that represents the collision box for the vehicle w.r.t. the camera.

5 Conclusion

In conclusion, the 3D object reconstruction and its pose estimation is computed using the above mentioned pipeline where the object reconstruction is shown in the form of 3D Pointcloud of the object calculated from the disparity of the 2D image. The bounding box enclosed on the object provides the pose of the object in 3D world coordinates which provides us with the information of border coordinates and the collision box for the object.

For future works, the results obtained can be used to predict the path for autonomous driving with respect to the location of different objects in the scene. Also, instance or panaoptic segmentation can be used instead of semantic segmentation for segmenting the object and considering various other classes. The 3D bounding box displayed on the Pointcloud can be transformed onto the 2D image to keep track of the objects in the camera frames rather than in Pointcloud data.

References

[1] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, C´eline Teuli‘ere, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2040–2049, 2017. 1, 2

[2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5410–5418, 2018. 7

[3] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2147–2156, 2016. 1, 2, 5, 6

[4] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In Advances in Neural Information Processing Systems, pages 424–432, 2015. 1, 2, 5, 6

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, page 3, 2017. 1, 3,