Thesis Proposal: Using a Modified Point Net to Generate 3D Bounding Box Proposals with Stereo Disparity Maps
April-August 2019

Student:	Kristian Gonzalez
	Ravensburg-Weingarten University of Applied Sciences
Supervisor:	Prof. Dr. Wolfgang Ertel
	Ravensburg-Weingarten University of Applied Sciences
Co-supervisor:	Prof. Dr. Stefan Elser
	Ravensburg-Weingarten University of Applied Sciences

1 Introduction

Computer vision is a field that has grown explosively in the last few years, very much in thanks to the utilization of convolutional neural networks, or CNN's, to accurately identify different kinds of information from raw image data. This capability has led to many different sub-fields of research, including 3D localization of objects using a variety of sensor setups. In having a branching set of possible sensor setups, a few questions naturally arise: how many sensors are needed to accurately locate objects, and how competitive are systems that do not directly take distance measurements to localize objects?

In the field of autonomous driving, a "typical" sensor system may include any combination of the following: forward facing camera, a complimentary second camera for stereo image generation, an IMU (inertial measurement unit), a lidar sensor (either a single sensor on the top or one at the front & back of the vehicle), a forward-facing radar, and possibly other cameras facing various directions. The KITTI dataset [7] features one lidar sensor with multiple cameras; the Oxford RobotCar [11], which focuses on driving through a similar area multiple times, especially contains two lidar sensors (front and rear); finally, some datasets such as Apolloscape [9] or Berkeley Deep Drive [16] obtain large amounts of RGB data, with less of a focus on lidar or stereo imaging.

Because of this sensor complexity, this paper proposes to seek out a more streamlined approach by primarily utilizing stereo vision to create 3D bounding box proposals and estimates. The current norm in the field leans heavily towards using lidar-based networks (typically nets that use some combination of camera and lidar information), while this paper instead asks: can stereo disparity maps provide a competitive alternative by using methods adapted from lidar networks? It must be acknowledged at the start of this proposal that lidar sensors do typically provide more accurate distance measurements by virtue of directly measuring the environment, but not without downsides. Lidar technology is typically expensive (although this price is decreasing), and has difficulty with reflective surfaces. Therefore, the aim of this paper is not to obtain better 3D object detection than lidar, but to demonstrate stereo vision's competitiveness.

To that end, this paper proposes a novel method of using stereo disparity maps to localize objects. A modified stereo depth estimation network, Pyramid Stereo Matching Network (PSMnet), will be used to generate a stereo image, which will then be projected to 3D point space and fed into a well-known point network, Frustum Point Net (FPnet).

2 Related Work

2.1 Convolutional Networks Using Point Clouds

A variety of networks exist that estimate 3-dimensional bounding boxes, including Frustum Pointnet, which is to be modified in this paper. FPnet takes a three step approach, which includes: "traditional" 2D object detection, 3D segmentation, and amodal estimation. This is demonstrated below in Figure 2.1, reproduced from the original paper.

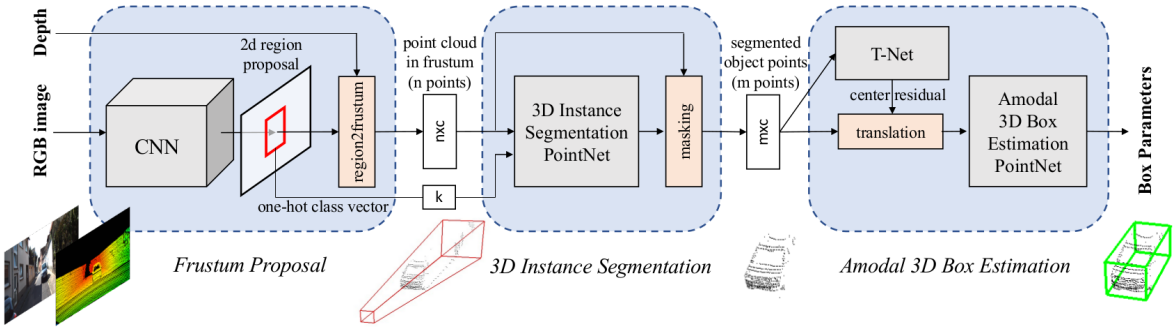


Figure 2.1: Reproduced image of general FPnet steps. First, 2D object detection locates an image and creates a "frustum" (or cone) of the valid lidar point cloud. Next, this reduced set of points segmented in the 3D space, by removing the fore- and background points. Finally, an estimate of the remaining points is made about the object's size and orientation to generate a bounding box.

The first point net to successfully implement this approach was the aptly-named "Pointnet" [13], which focused on working with point cloud data in its native format, rather than transform the data to 3D voxel grids or views. This approach provides some clear advantages over networks like Voxnet [12], which relies on creating a three-dimensional occupancy grid requiring a reference frame and given resolution. Though creating volumetric or 3D CNN's created an initial approach, they are limited by data sparsity and computation cost. By contrast, the original Pointnet and subsequent FPnet are designed to work with a point cloud's inherent properties, paraphrased from [13]:

- Unordered: point clouds are essentially an unordered set of vectors containing x,y,z coordinates, among other information.
- Point interaction: because the points come from euclidean space, neighboring points form a meaningful group, so any network must be able to represent the "local structures" of a subset of points.
- Invariance under transformation: some simple transformations, such as rotation, translation, or scaling, do not change the inherent category the object belongs to, such as a car or plane.

Using multiple views is another approach taken to tackle point cloud data. In practice, multi-view CNN's project 3D point clouds onto 2D planes, then apply 2D CNN's for classification. However, this rapidly increases in complexity when extending the usage to understanding or shape completion.

2.2 Using CNN's to Generate Stereo Disparity Maps

Creating a disparity map by use of two cameras, or stereo matching,has its roots in finding a matching pixel between two cameras to understand its distance away from the cameras by use of epipolar

geometry. Typically, an image similar to Figure 2.2 (reproduced from Hamzah and Ibrahim [8]) below is used to illustrate this.

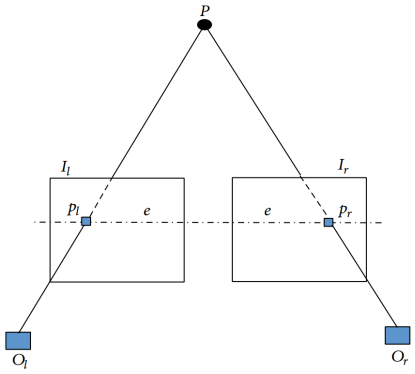


Figure 2.2: Using trigonometry, a target at point P can be located on two image planes at their local points and estimated at a certain distance from the sensors.

Ibraim and Hamzah [8] present a comprehensive but not exhaustive survey on existing algorithms, several of which used only a CPU for their calculations. However, as also stated, the "number of calculations required increases with an increasing number of pixels", making the matching problem computationally complex. Of course, any discussion of stereo correspondence algorithms warrants mention of the four classical steps of stereo matching, per [15]: matching cost computation, aggregation, optimization, and refinement.

To know the state of the art of stereo disparity map creation, one may look at various benchmarks that use a single, common dataset to compare multiple networks and algorithms. The well-known Middlebury [14, 3] dataset is a common benchmark, as well as the KITTI [7, 2] 2015 Stereo benchmark. Additionally, there are even some aggregate lists, including the 2018 Robust Vision Challenge [4] leaderboard. Looking at the aggregated Robust Vision Challenge leaderboard, last updated in December 2018, two top-performing networks are iResNet, holding 1st place, as well as PSMnet (Pyramid Stereo Matching network), in 4th place. What also makes these two stereo networks special is that both have publicly available repositories. Furthermore, PSMnet, or Pyramid Stereo-Matching Network, has simpler and fewer dependencies than iResNet. Lastly, PSMnet is (as of this writing) in the top 10 of the current KITTI 2015 stereo vision challenge leaderboard.

PSMnet itself is comprised of two main modules, as stated by the authors: spatial pyramid pooling and a 3D CNN. The overall architecture, reproduced from the original paper, is shown below in Figure 2.2.

An interesting note on the spatial pyramid pooling module is how it addresses fixed-size constraint in CNN's. At various dimensional sizes, feature maps are generated and flattened, which are then fed into a fully connected classification layer. Finally, these features are compressed via adaptive average pooling, then upsampled back to the original feature map size, leaving a multi-level feature map for consumption of the 3D CNN.

2.3 Using Stereo Disparity Maps and CNN's for 3D Object Detection

There is a relatively small body of literature on stereo-based 3D object detection. As of the time of this writing, Li et al. [10] created the only stereo-based network on the 3D KITTI Vision Benchmark,

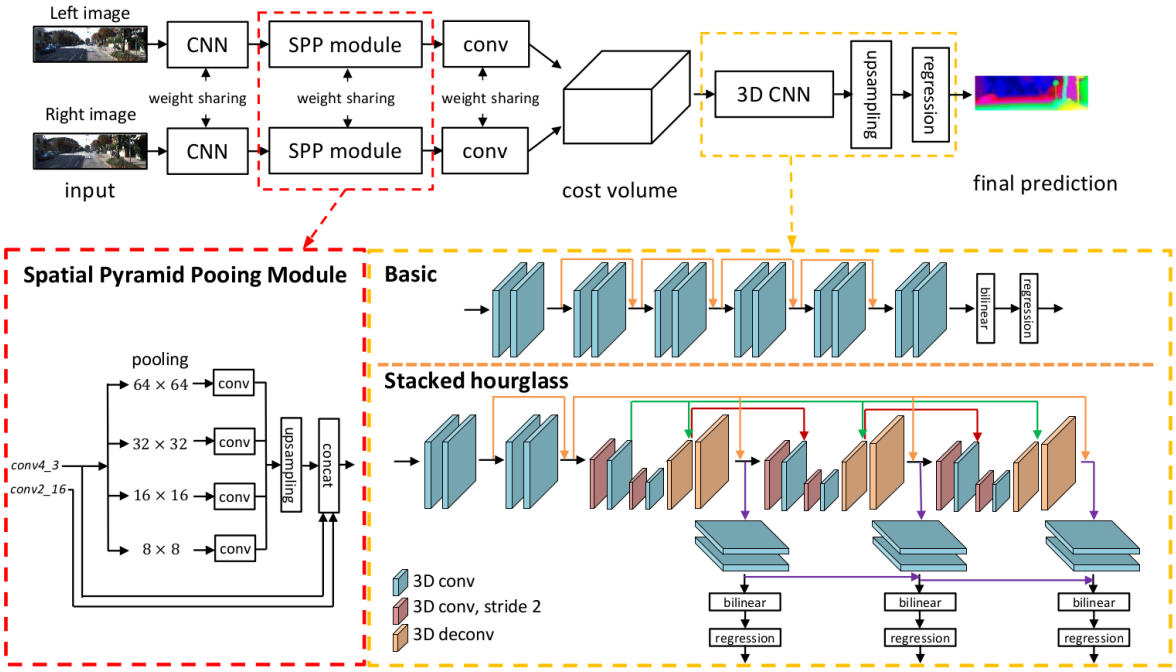


Figure 2.3: Reproduced from original paper [5]. Two main components, spatial pyramid pooling (harvests features) and a 3D CNN (regularizes cost volume / performs disparity regression) form the basis of the network.

Stereo R-CNN. This network is inspired by Faster R-CNN and also features weight sharing and a region proposal network. What sets this network apart is how it predicts object keypoints, which are then used in 3D box estimation. Each image stream has a stereo RPN (region proposal network with shared weights, which is then processed with an ROI Align operation; each ROI pair is then used to generate the relevant keypoints via "photometric alignment", or more specifically the left region of interest is warped to reduce the photometric error, leading to find the "best center depth".

Chen and Kundu [6] make a primary assumption that all relevant classes are on the ground plane, and "only depth information (no appearance)" is used in the 3D object proposal generation process. Additionally, each object proposal is taken from a set of templates, learned in the training data to represent the typical size of each class, requiring some a priori information.

3 Proposed Solution

Given the current state of 3D localization using stereo vision, as well the opportunity it presents, this paper proposes to investigate taking a new approach on 3D localization using stereo data. The key idea is to use inspiration from Frustum Pointnet, taking a point cloud and working with that data in 3D space rather than 2D space, and extend this to stereo data. Using an appropriate transform, disparity data that is projected onto the 2D image plane can be moved to 3D space, and treated as a point cloud. With this pseudo-point cloud, generated from a best-in-class stereo algorithm such as Pyramid Stereo Matching, Frustum Pointnet will be trained and evaluated on its performance with this new data input. Naturally, there are several moving components in this idea, and each must be addressed:

- Obtaining stereo disparity maps from PSMnet

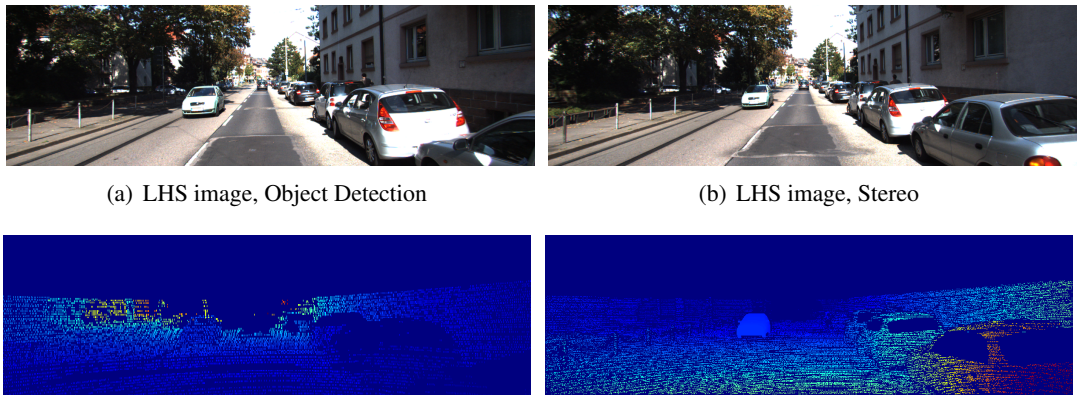
- Transforming between lidar data and stereo data
- Feeding this data into FPnet
- Evaluating performance on 3D bounding boxes
- Overall Project Timeline

3.1 Obtaining stereo disparity maps

In order to obtain stereo disparity maps, Pyramid Stereo Matching has been selected to output this data. The networks is easily one of the best in its class, and has openly available code. In fact, this network has already been downloaded, configured, and modularized as of the time of this writing. This means that the important first step of obtaining stereo images from the dataset has been completed. Additionally, this means that the network has been simplified down to a simple object that can be called: the loaded model is given a stereo image pair and returns an estimated disparity map. The model was trained following the instructions by the original authors: first it was trained from scratch on Freiburg SceneFlow data for 10 epochs, then finetuned on KITTI 2015 stereo data for 300 epochs. This has thus produced a ready-to-use solution for stereo imaging using the KITTI dataset.

3.2 Transforming between Lidar Data and Stereo Data

The task of transforming stereo data into a point cloud (and perhaps the reverse as well) is also underway, and already has some initial results. There are some important steps required to transform a stereo image into something that can imitate a point cloud, but a side-by-side comparison of two similar images may provide some insight. There are technically two datasets being demonstrated below from KITTI benchmark, the stereo 2015 dataset and the 3D object detection dataset. As seen below in figure 3.2, there are some visual correlations between a transformed image (c) and the stereo ground truth (d).



(a) LHS image, Object Detection

(b) LHS image, Stereo

(c) Projected Lidar, Object Detection

(d) Disparity Ground Truth, Stereo

Figure 3.1: Side-by-side comparison of similar images from different datasets and their respective depth images. Sub-figure (c) was created by taking the record's lidar scan and projecting the points onto the image plane. By contrast, sub-figure (d) is the stereo dataset record's ground truth image, and seems to have higher values for points that are closer.

This aspect of the project, while important, is expected to take some time to develop a suitable transformation function, but should feature simpler mathematics by comparison. The main task is

not to identically match each pixel that lidar and stereo data share, but rather to reduce error as much as possible so that the difference between stereo and lidar performance is not due to the conversion between one format to the other.

3.3 Feeding Data into FPnet

In order to modify the data going into Frustum Pointnet, there must be some understanding of how to interact with the original data. This step may take some iteration, but is expected to be understood within the first half of the thesis. The original author's repository for FPnet uses Tensorflow, different than the PyTorch implementation of Stereo Pyramid Matching. This means that there will generally be some slowdown when converting from one format to another, but there is no back-propagation that will be performed from one network to the other. The primary idea here is that FPnet will receive either a stereo disparity map or a pseudo-point cloud that comes from a previously trained, fully functioning stereo generation network. Thus, FPnet will take the data as-is, and learn to estimate distance from these values.

3.4 Evaluating Performance on 3D Bounding Boxes

In order to properly evaluate the detections of the network's 3D bounding boxes, the well known methods of evaluating 2D bounding boxes must be adapted. This has already been investigated and implemented by David Stutz and Bo Li [1]. Thus, this method will be followed to evaluate and compare performance of the network. If other, more accurate methods become available in the course of the project, they will be used instead.

3.5 Project Timeline

In order to meet all requirements while also satisfying educational program requirements, a proposed timeline is provided below. Notable dates here include the official start of the thesis, the final date of the thesis, including presentation and document turn-in.

References

[1] The kitti vision benchamark suite, 3d object detection evaluation 2017. `http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d`. Accessed: 15.03.2019.

[2] The kitti vision benchamark suite, stereo evaluation 2015. `http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo`. Accessed: 06.03.2019.

[3] Middlebury stereo evaluation - version 3. `http://vision.middlebury.edu/stereo/eval3/`. Accessed: 06.03.2019.

[4] Robust vision challenge 2018. `http://www.robustvision.net/leaderboard.php?benchmark=stereo`. Accessed: 06.03.2019.

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid Stereo Matching Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018.

	Week No. (#) / Week Start (DD-MM)	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
		04	11	18	25	01	08	15	22	29	06	13	20	27	03	10	17	24	01	08	15	22	29	05	12	19	26
No.	**Task Name**	03	03	03	03	04	04	04	04	04	05	05	05	05	06	06	06	06	07	07	07	07	07	08	08	08	08
0.0	Literature Search	1	2																								
0.1	Approval / Submission of Project			1																							
1.0	Collection / Formatting of dataset			1																							
2.0	Configure stereo network (able to train, test)	1	2																								
2.1	Modularize stereo network (use as library)		1	2																							
2.2	Setup stereo-point cloud conversion	1	2	3	4																						
3.0	Configure point network (able to train, test)			1	2	3	4																				
4.0	Modify point network to accept stereo data						1	2	3	4	5	6	7	8	9	10											
4.1	Train point network on new data										1	2	3	4	5	6	7	8									
4.2	Test, evaluate point network on new data														1	2	3	4	5	6	7	8	9				
4.3	Obtain results for modified point network																				1	2	3				
5.0	Thesis first draft creation				1	2	3	4	5	6	7	8															
5.1	Thesis first draft review (50% done)												1														
5.2	Thesis second draft creation													1	2	3	4	5	6	7	8	9					
5.3	Thesis second draft review (80% done)																						1				
5.4	Thesis final draft creation																							1	2		
5.5	Thesis final draft review (99% done)																									1	
5.6	Thesis submission																										1
5.7	Presentation submission																										1

Figure 3.2: Proposed schedule of paper. Highlighted tasks are critical milestones. Numbers inside of gray cells indicate the number of weeks each individual task lasts.

[6] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *CoRR*, abs/1608.07711, 2016.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] Rostam Affendi Hamzah and Haidi Ibrahim. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016, 2016.

[9] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The ApolloScape Dataset for Autonomous Driving. *arXiv: 1803.06184*, 2018.

[10] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. *arXiv preprint arXiv:1902.09738*, 2019.

[11] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[12] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015.

[13] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[14] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.

[15] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[16] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100k: A Diverse Driving Video Database with Scalable Annotation Tooling. *CoRR*, abs/1805.04687, 2018.