LAPTNet: LiDAR-Aided Perspective Transform Network

Manuel Alejandro Diaz-Zapata, Özgür Erkent, Christian Laugier, Jilles Dibangoye, David Sierra González

LAPTNet: LiDAR-Aided Perspective Transform Network

Manuel Diaz-Zapata[1,2], Özgür Erkent[1,3], Christian Laugier[1], Jilles Dibangoye[1,2], David Sierra-Gonzalez[1]

***Abstract*— Semantic grids are a useful representation of the environment around a robot. They can be used in autonomous vehicles to concisely represent the scene around the car, capturing vital information for downstream tasks like navigation or collision assessment. Information from different sensors can be used to generate these grids. Some methods rely only on RGB images, whereas others choose to incorporate information from other sensors, such as radar or LiDAR. In this paper, we present an architecture that fuses LiDAR and camera information to generate semantic grids. By using the 3D information from a LiDAR point cloud, the LiDAR-Aided Perspective Transform Network (LAPTNet) is able to associate features in the camera plane to the bird's eye view without having to predict any depth information about the scene. Compared to state-of-the-art camera-only methods, LAPTNet achieves an improvement of up to 8.8 points (or 38.13%) over state-of-art competing approaches for the classes proposed in the NuScenes dataset validation split.**

I. Introduction

For an autonomous vehicle, sensing its surroundings is a crucial task. To do this, the vehicle can make use of an array of different sensors, such as cameras or LiDARs, to gather information about the environment that it is in. Cameras have been one of the most widely used sensors for tasks such as image segmentation, 2D and 3D object detection [1], [2], [3]. LiDARs have also been used to a lower extent for 3D point cloud segmentation [4] and 3D object detection [5]. Other works have also explored the fusion of both types of sensors for tasks like detection with great success [6].

Many methods have used a camera as the only sensor. They usually address tasks that are performed in the camera plane, like semantic segmentation [1] or object detection [8]. These works, although important to the field of scene understanding, rely on a representation space that suffers from perspective distortions [9]. This can result in differences of the final output for objects with similar sizes in the real world, e.g. bigger bounding boxes or segmentation masks for cars closer to the autonomous vehicle than for those farther away.

With multi-layer LiDAR sensors, perspective distortion is not an issue since the points are already in 3D. Here, the problem is related to the density of the information and the structure in which they will be processed. Compared to a camera, the amount of data points given by LiDAR is very sparse, which can result in the loss of important details present in the scene.

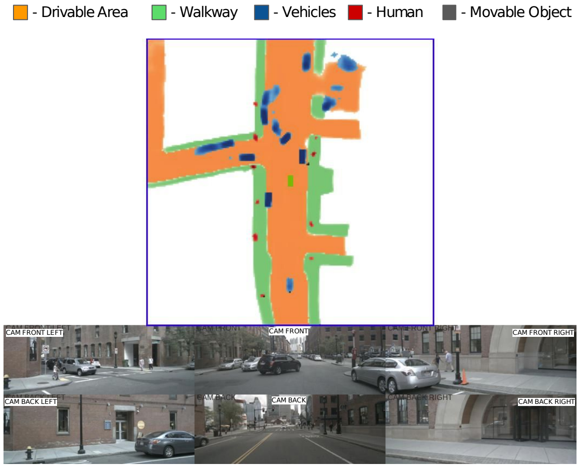

Fig. 1: Semantic grid predictions by LAPTNet for classes in the NuScenes dataset [7]. By using LIDAR information to aid the projection of camera features, our method allows the creation of more precise semantic grids compared to camera-only methods. Best viewed with digital zoom.

In robotics, occupancy grids are a representation space that can be quickly generated from 3D information [10]. They usually represent a discretized version of the top-down view, sometimes called bird's-eye view (BEV), which is usually the plane in which a robot can move. For this, a 3D point cloud can be used to generate a 2D array that indicates which cells are occupied or not. In comparison, using a monocular image to build this representation space using only geometrical methods, like inverse perspective mapping [9], is very challenging. This is due to the uncertainty of the pinhole model for determining the depth correspondence for each pixel in the image [9]. This problem can be alleviated by using more than one camera at the same time, as in stereo depth estimation, or by using feature matching methods for a moving camera, as in structure from motion [9].

Recently, the problem of how to generate semantic grids has started to receive lots of attention from the community working on perception for autonomous vehicles [11], [12], [13]. Here, the sensor that has received most of the attention has been the camera. Semantic occupancy grids have been estimated by leveraging stereo depth [14] or by using neural networks to learn implicit depth distributions [15], to go from the camera plane to the BEV plane. The fusion of camera and LiDAR information to generate semantic grids has also been explored by other works [11], [16], [17] using different approaches.

In this paper, we propose a fusion architecture that leverages the 3D information from LiDAR together with the

This work was partially supported by EU project CPS4EU and Toyota Motor Europe.

[1] Authors are with the Chroma team, INRIA Grenoble Rhone-Alpes, France. [2] Authors are with CITI-Lab, INSA Lyon, France. [3] Author is with Hacettepe University, Ankara, Turkey. Correspondence: `manuel.diaz-zapata@inria.fr`

features extracted from camera images to generate semantic grids. The proposed approach uses the information from the point cloud to guide the projection of features extracted from camera images, performing the association between the image plane and the bird's-eye view representation in a fast and efficient manner without having to rely on estimation methods to project image information to the BEV. We address the sparse association problem between camera pixels and LiDAR points by leveraging the downsampling feature from convolutional neural networks (CNNs) to do the point-pixel correlation. In fig. 1, the resulting semantic grids can be seen with the corresponding set of surround images used to generate them.

In Section II, we present the related literature; in Section III, we explain our network architecture, and how LiDAR information helps to project the image features onto the BEV to generate semantic grids; in Section IV, we describe our experimental setup on the NuScenes Dataset and in Section V we present our results on the NuScenes dataset with 5 different classes. Finally, we summarize the findings of our work.

II. Related Work

In this section we will present some of the current works in the literature which use the bird's eye view space for different perception tasks and how it is generated from camera or LiDAR data. We will begin by presenting some works on 3D bounding box detection in subsection II-A, followed by methods that estimate semantic grids through camera-based methods, LiDAR-based methods and fusion-based methods in subsection II-B.

A. Using the Bird's Eye View for bounding box detection

Recent works have been interested in using the BEV space as an intermediary representation space for the prediction of 3D and 2D bounding boxes from either camera or LiDAR information. Camera-only methods perform the projection from the camera plane to the BEV using different approaches. OFT uses an intermediary voxel space to associate image features to the BEV [18]. A Generative Adversarial Network [19] is used to do the projection in [20]. In CaDDN, the camera features are projected to 3D via categorical depth distributions that are predicted with a Frustum Feature Network, they are later collapsed to the BEV via convolutions [3].

Aside from camera-based models, other methods like Voxel R-CNN use LiDAR, where a BEV representation is generated by stacking in the Z axis the voxels generated from the point cloud that are processed by a 3D network [21]. Deep Continous Fusion [22] proposes to use the BEV as the space to fuse information from cameras and LiDAR through a 'Continous Fusion Layer' based on [23].

B. Semantic Grid estimation

The bird's eye view representation space can be better related to the task of estimating semantic grids than the estimation of 3D bounding boxes. To create them, cameras and LiDARs are two of the most used sensors. In this subsection, will present how semantic grids have been created using only camera or LiDAR data, as well as some sensor fusion approaches.

1) Using only cameras: In the Pyramid Occupancy Network the proposed dense transformers project each of the output scales from an FPN [24]-inspired backbone to the feature map in the BEV [12]. Lift-splat-shoot tackles the projection problem by jointly predicting a set of categorical depths together with the image features from a 2D backbone, then performing voxel pooling to generate the BEV used for predicting the semantic grid [15] . In the VPN, a set of camera viewpoints are projected to the BEV using a View Transformer Module, which is a multilayer perceptron that finds the relationship between each pixel in each view to each cell in the BEV [25]. Yang *et al.* use attention mechanisms to project extracted features from monocular images to the BEV, similar to the VPN [26]. Hoyer *et al.* project semantic labels from the image plane to the BEV using stereo depth information [14].

For the camera-base approaches, the pinhole model does not allow to project from the 2D image to a certain 3D point without extra depth information. These methods address this shortcoming in different ways by using stereo depth estimation, learning depth distributions or by using an intermediary bottleneck dimension to go from the camera plane to BEV. These estimation approaches present inherent uncertainties that affect the precision of the models for the creation of semantic grids.

2) Using only LiDAR: PillarSegNet [13] extends the work of [5] to generate semantic grids by using a combination of Pillar features with an occupancy feature map (both in the BEV). CMCDOT [10] estimates static and dynamic occupancy states, as well as empty and unknown states for cells in a grid using a Bayesian filtering approach.

Although LiDAR approaches have the advantage of working with 3D information from point clouds, they do not offer the density of information, nor the texture information available in RGB images that can be beneficial to understand how the scene is composed.

3) Fusing different sensors: Erkent *et al.* propose the late fusion of camera and LiDAR data on the BEV through an encoder-decoder network to predict the semantic grid. Semantic segmentation images are projected via inverse perspective mapping to a set of intermediary planes which are then concatenated to CMCDOT grids to be fused together [27]. FISHINGNet does late fusion for information from cameras, LiDAR and radar to predict semantic grids. Camera features are projected here using the VPN [25]. Each modality is processed separately to predict semantic grids, they are then fused via pooling to generate the final output [28].

These methods rely on camera-based approaches for the projection of image features to the BEV, as well as separate pipelines for LiDAR and camera data. An early fusion scheme could allow sensors to cover each others shortcomings earlier, and their information to be jointly processed for faster inference.

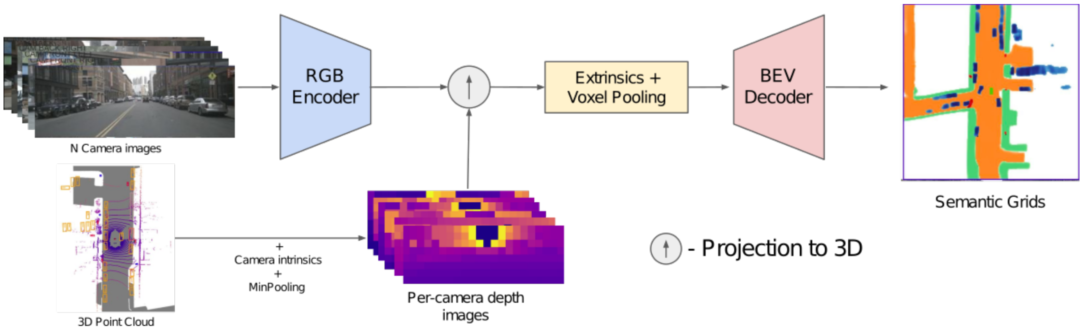

Fig. 2: Proposed architecture for the LAPTNet. We encode per-camera features using a CNN on the images. These features are then projected to 3D using their corresponding depth images. These depth images are generated by projecting the LiDAR point cloud to each of the cameras' field of view using the intrinsics of the cameras together with a minpooling operation to match the downsampling performed by the CNN. The points in 3D are aligned to the vehicle's reference frame and projected to the BEV through a voxel sum-pooling operation [15]. Finally, a BEV decoder network generates the semantic grids from this BEV feature map.

III. LAPTNet: LIDAR-Aided Perspective Transform Network

In this section we describe how our approach, the LIDAR-Aided Perspective Transform Network (LAPTNet) generates semantic grids by using 3D point clouds to guide the projection of camera features to the BEV. A general overview of the method can be seen in Fig. 2.

A. Problem formulation

We are given n images $\{\mathbf{X}_k \in \mathbb{R}^{3\times H\times W}\}_n$ taken from the cameras located around a vehicle, each with a corresponding intrinsic ($\mathbf{I}_k \in \mathbb{R}^{3\times 3}$) and extrinsic ($\mathbf{E_k} \in \mathbb{R}^{4\times 4}$) camera matrix. We are also given a point cloud ($\mathbf{P} \in \mathbb{R}^{3\times D}$) taken by the LiDAR at the same time, with its corresponding transformation matrix ($\mathbf{E}_P \in \mathbb{R}^{4\times 4}$) from the vehicle's reference frame.

Using this information, we want to estimate an occupancy grid ($\mathbf{y} \in \mathbb{R}^{C\times X\times Y}$) in the BEV centered on the coordinate frame of the vehicle. Here, by leveraging the 3D geometric information available in $\mathbf{P}$, we can project features from $\mathbf{X}_k$ to the BEV without having to deal with the depth estimation task for each pixel.

B. Processing of camera images

We want to have as many correspondences between points in P and pixels in $\mathbf{X}_k$ to generate a representation in the BEV. But given the sparsity of data in point clouds compared to the pixel density in images, if we directly project the point cloud to the image, a big amount of information will be lost since not all pixels in the original image will get a point correspondence to perform the projection to the BEV.

Instead of projecting the RGB values from the original image, we choose to preprocess $\mathbf{X}_k$ with CNNs. Following the standard CNN encoding, we are able to downsample each $\mathbf{X}_k$ by a factor of d_f to a feature map $\{\mathbf{F}_k \in \mathbb{R}^{N_f\times H/d_f\times W/d_f}\}_n$ containing N_f channel-wise features. This allows us to perform the association in a smaller space with the possibility of finding a higher ratio of point-pixel correspondences.

C. Projection of camera features onto the BEV plane

The main idea behind LAPTNet is to leverage the geometrical information about the scene, encoded in $\mathbf{P}$, to associate the pixels in $\mathbf{X}_k$ from the camera plane to their corresponding cells in the intermediary BEV representation ($\mathbf{B} \in \mathbb{R}^{N_f\times X\times Y}$). By projecting $\mathbf{P}$ to the field of view of $\mathbf{X}_k$ we can associate image features with their corresponding point in 3D where each pixel is looking to.

For this, we transform $\mathbf{P}$ from the LiDAR reference frame to the camera reference frame using the camera's extrinsics ($\mathbf{E}_k$) and the LiDAR transformation matrix ($\mathbf{E}_P$). The transformation is done using homogeneous coordinates as indicated in equation 1.

$$\begin{pmatrix}\mathbf{P}_k\\ 1\end{pmatrix} = \mathbf{E}_k \times \mathbf{E}_P^{-1} \times \begin{pmatrix}\mathbf{P}\\ 1\end{pmatrix} \tag{1}$$

With the points now in the camera reference frame ($\mathbf{P}_k = (x_k, y_k, z_k)^T$), a perspective transformation from 3D space to the 2D image coordinates (u_k, v_k) is performed using the camera's intrinsic matrix ($\mathbf{I}_k$). As can be seen in equation 2, we normalize by the value in the depth dimension (z_k) for each point in $\mathbf{P}_k$.

$$\begin{pmatrix}u_k\\ v_k\\ 1\end{pmatrix} = \mathbf{I}_k \times \begin{pmatrix}x_k/z_k\\ y_k/z_k\\ z_k/z_k\end{pmatrix} \tag{2}$$

Knowing where the points from $\mathbf{P}$ are projected to $\mathbf{X}_k$, we only keep the point closest to the camera coordinate frame in the z_k axis. We choose the closest point since we are using the pinhole camera projection model [9]. With this information, we create a sparse depth image $\{\mathbf{D}_k \in \mathbb{R}^{1\times H\times W}\}_n$ for each camera, saving the depth as the pixel value where the points are projected to.

Finally, in order to use the information in $\mathbf{D}_k$ to project $\mathbf{F}_k$ to the BEV, we need to reduce the depth image's dimensions to match the feature map. By performing a minpooling operation with a kernel of size d_f we find the closest distance value in the receptive field for each pixel of $\mathbf{F}_k$ in $\mathbf{D}_k$. An example of this low-resolution depth map can be seen in Fig.

	Human	Vehicle	Movable Object	Drivable Area	Walkway
VPN [25]	7.1%	13.47%	7.7%	58.0%	29.4%
PON [12]	8.2%	15.37%	6.9%	60.4%	31.0%
Lift-Splat-Shoot [15]	9.99%	32.02%	21.6%	77.6%	51.03%
FISHINGNet (LiDAR and Camera) [28]	**20.4%**	**40.9%**	-	-	-
LAPTNet (Ours)	13.8%	40.13%	**27.45%**	**79.43%**	**57.25%**

TABLE I: Results on the NuScenes validation split. We perform the comparison of the Intersection over Union of the generated semantic grids. Best results are presented in bold font.

	Human	Vehicle	Movable Object	Drivable Area	Walkway
Lift-Splat-Shoot (Rain)	5.16%	33.2%	27.15%	71.95%	47.05%
LAPTNet (Rain)	**10.05%**	**44.76%**	**32.37%**	**73.07%**	**49.96%**
Lift-Splat-Shoot (Night)	4.99%	31.44%	4.85%	64.47%	22.93%
LAPTNet (Night)	**6.29%**	**36.8%**	**12.3%**	**67.88%**	**25.9%**

TABLE II: Results on the NuScenes validation scenes under rain (top) and night (bottom) conditions.

2. These distance values $\{\delta_k \in \mathbb{R}^{1\times H/d_f \times W/d_f}\}_n$ are used to do the projection to 3D in the camera's reference frame as shown in equation 3. A feature located in coordinates $(u_f, v_f)^T$ in $\mathbf{F}_k$, will be projected to the point $(x_f, y_f, z_f)^T$ in 3D space.

$$\begin{pmatrix} x_f \\ y_f \\ z_f \end{pmatrix}_k = \mathbf{I}_k^{-1} \times \delta_k \times \begin{pmatrix} u_f \\ v_f \\ 1 \end{pmatrix}_k \tag{3}$$

$$\begin{pmatrix} x_f \\ y_f \\ z_f \\ 1 \end{pmatrix}_{car} = \mathbf{E}_k^{-1} \times \begin{pmatrix} x_f \\ y_f \\ z_f \\ 1 \end{pmatrix}_k \tag{4}$$

With the point cloud containing the features' position in 3D, we transform them from the camera's reference frame to the vehicle's reference frame (equation 4). Then, we perform the projection to the BEV by following the voxel pooling method described in [15]. We create the intermediary representation $\mathbf{B}$ by assigning every point $(x_f, y_f, z_f)^T_{car}$ to its nearest pillar and performing sum pooling in each pillar. Here, "pillars" refer to voxels with infinite height as described in [5].

D. Semantic grid generation using the projected features

Since the representation space $\mathbf{B}$ follows the same structure as an image, we use a lightweight CNN as the BEV decoder that outputs the semantic grid. This top-down network is a ResNet-18 network that ends with 2 upsampling blocks (bilinear upsampling operation, (3x3) convolution, batch normalization and ReLU) to recover the original grid spatial size (X, Y) after passing through the decoder. A final (1x1) convolution predicts the wanted output ($\mathbf{y} \in \mathbb{R}^{C\times X \times Y}$) with C channels for each cell in the grid.

IV. Experimental Setup

In this section we will discuss the experimental setup for the training of LAPTNet. We will describe the dataset, loss function and metric used for the evaluation of model performance under general and difficult conditions.

A. Dataset

We base all of our experiments on the NuScenes dataset [7]. NuScenes is a large dataset with 1000 driving scenes from different locations around the world. Each scene of the dataset has a duration of 20 seconds, recording information from a variety of sensors such as LiDAR, cameras and radars. Since the semantic grid ground truth is not directly available from the dataset, we extend the method of [15] to generate the ground truth for all of the chosen classes. Using the 3D bounding box annotations and the high-definition maps available in the dataset, we generate our ground truth semantic grids for the different classes. Given the annotation scheme followed by NuScenes, we decide to group the 3D bounding box annotations into the classes 'human', 'movable object' and 'vehicle' as well as taking the 'drivable area' and 'walkway' classes from the HD map.

B. Loss function and Evaluation metrics

We train our network separately for each class ($C = 1$). Knowing this, we employ the binary cross entropy loss as our loss function with a weight for positive samples equal to 2.13. We chose this value given the implementation of [15]. We use the Intersection over Union (IoU) metric to evaluate how similar the predicted segmentation masks are to the ground truth.

C. Competing approaches

To the extent of our knowledge, no other method apart from [28] generates semantic grids using a sensor fusion approach in the NuScenes dataset. Knowing this, we compare our method to current state of the art baselines that use only camera-based approaches such as [15], [12], [25].

We also report the performance of our model under specific conditions such as those of rain and night. These two conditions are of interest to us given the effect that rain can have in the accuracy of LiDAR as well as low-light for cameras.

V. Results

In this section we discuss the results of our approach, how it compares to some of the current state of the art methods

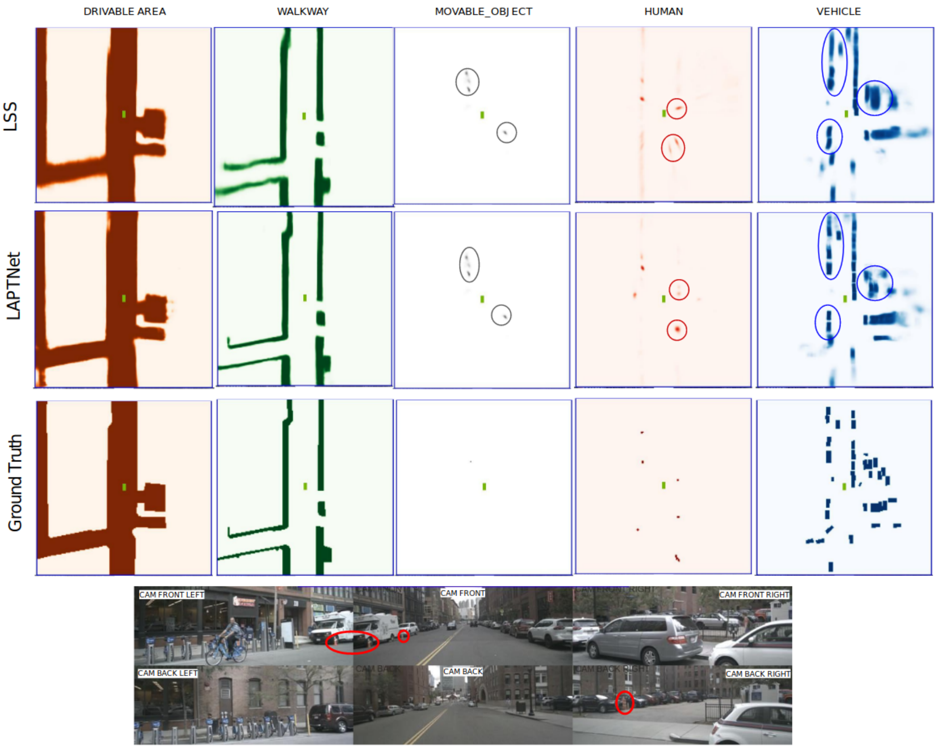

Fig. 3: Comparison of the predicted semantic grids by Lift-Splat-Shoot (LSS) [15] and LAPTNet. For some of the classes whose objects occupy a few cells in the grid, like human or movable object, LAPTNet seems to predict a more precise location than LSS. The red circles in the images highlight some movable objects that are not present on the ground truth annotations and are detected by both methods (third column). For classes such as vehicles, movable objects and humans the predictions from LSS are stretched along the projection ray, this does not happen in the LAPTNet predictions. Best viewed with digital zoom.

and how our method compares under conditions such as night and rain.

A. *Comparison against state of the art*

We report our quantitative results on Table I. In this table, we can observe that by adding the LiDAR information we are able to outperform all of the camera-only baselines. We hypothesize that this improvement is due to the addition of the LiDAR data, since the real depth information available in it informs where objects are located in space better than prediction-based baselines.

We report improvements of **3.8** points (38.13%) in the human metaclass, of **8.8** points (25.33%) in the vehicle metaclass, of **5.85** points (27.08%) in the movable object metaclass, of **1.83** points (2.35%) in the drivable area class and of **6.22** points (12.18%) in the walkway class over the highest performing camera-only method (LSS [15]).

We also observe that even without adding a LiDAR-specific encoder, we get close results to FISHINGNet [28] for the vehicle class. A noticeable difference can be seen in the performance for the human class, showing potential for improving the performance of LAPTNet. By adding a feature extractor specific for LiDAR to be fused with the projected camera features, we expect to improve the performance of the network. We also interpret these results as another indicator of the overall value generated from adding the real 3D information from the LiDAR sensor.

A sample of the qualitative results compared against the best competing method can be seen in Fig. 3. Here it can be seen that LAPTNet manages a more concise assignment of cells that belong to classes with a smaller footprint such as human and movable object. In contrast for LSS, the prediction masks are streched along the projection ray used to reach those cells. For vehicles, LAPTNet is able to more precisely assign the cells to which the vehicles belong to. We also note the interesting case for the two traffic cones highlighted by a red circle in the camera images of Fig. 3, where even if they are not annotated in the ground truth, both LAPTNet and LSS seem to have detected them.

B. *Performance under difficult conditions*

Since we do not rely on only one type of sensor for our prediction, we expect our method to be robust under adverse conditions such as rain or night. The results for the scenes under rain condition can be seen in the upper part of table II and the results for night condition can be seen in the lower part.

Comparing against the highest-performing camera-based model, we observe that our method outperforms the baseline in both types of adverse conditions. In this study we find further support to the idea that not only the color or texture features taken from the camera images are important to generate the semantic grids, but the geometric distribution of them within the grid space (given by the LiDAR) is what enables our method to outperform the baselines.

Looking at the results for both rain and night, we think that our method still seems to rely more on the camera features than the geometric distribution of them along the grid. This idea stems from the fact that cameras perform worse under low-light situations, and since the projected features in the BEV come from a camera-only encoder, even adding the depth information from the LiDAR is not enough to fully overcome this challenge. On the other hand, when comparing the performance under rainy conditions, a bigger challenge for LiDARs than for cameras, we see that the drop in segmentation accuracy is not as big as it is for night conditions. We have reason to think that, as previously stated, adding a parallel branch in the style of [5] can help us improve the results in the night condition since LiDAR is not affected by this type of adversity.

VI. Conclusion and Future Work

We have presented here the novel approach of the LIDAR-Aided Projective Transform Network (LAPTNet). This network uses real geometric information from a LiDAR sensor to guide the projection of camera features onto a bird's eye view perspective for semantic occupancy grid generation. Our method consistently improves the grid segmentation performance in the classes defined for the nuScenes dataset.

As future work, we plan to evaluate how the performance of the model changes when using multiple feature scales. We also plan to evaluate if the performance changes when the LiDAR information is processed with a separate encoder.

Acknowledgement

The experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations.

References

[1] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *CVPR*, 2017.

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8555–8564.

[4] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[5] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[6] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918–927.

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[8] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[9] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[10] L. Rummelhard, A. Nègre, and C. Laugier, "Conditional monte carlo dense occupancy tracker," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2485–2490.

[11] O. Erkent and C. Laugier, "Semantic segmentation with unsupervised domain adaptation under varying weather conditions for autonomous vehicles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3580–3587, 2020.

[12] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 138–11 147.

[13] J. Fei, K. Peng, P. Heidenreich, F. Bieder, and C. Stiller, "Pillarsegnet: Pillar-based semantic grid map estimation using sparse lidar data," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 838–844.

[14] L. Hoyer, P. Kesper, A. Khoreva, and V. Fischer, "Short-term prediction and multi-camera fusion on semantic grids," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[15] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *European Conference on Computer Vision*. Springer, 2020, pp. 194–210.

[16] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le, *et al.*, "Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 182–17 191.

[17] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[18] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," *British Machine Vision Conference*, 2019.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[20] S. Srivastava, F. Jurie, and G. Sharma, "Learning 2d to 3d lifting for object detection in 3d for autonomous vehicles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4504–4511.

[21] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," *arXiv preprint arXiv:2012.15712*, vol. 1, no. 2, p. 4, 2020.

[22] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 641–656.

[23] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2589–2597.

[24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[25] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4867–4873, 2020.

[26] W. Yang, Q. Li, W. Liu, Y. Yu, Y. Ma, S. He, and J. Pan, "Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 536–15 545.

[27] Ö. Erkent, C. Wolf, and C. Laugier, "End-to-end learning of semantic grid estimation deep neural network with occupancy grids," *Unmanned systems*, vol. 7, no. 03, pp. 171–181, 2019.

[28] N. Hendy, C. Sloan, F. Tian, P. Duan, N. Charchut, Y. Xie, C. Wang, and J. Philbin, "Fishing net: Future inference of semantic heatmaps in grids," *arXiv preprint arXiv:2006.09917*, 2020.