TransFuseGrid: Transformer-based Lidar-RGB fusion for semantic grid prediction

Gustavo Salazar-Gomez, David Sierra González, Manuel Alejandro Diaz-Zapata, Anshul Paigwar, Wenqian Liu, Özgür Erkent, Christian Laugier

TransFuseGrid: Transformer-based Lidar-RGB fusion for semantic grid prediction

Gustavo Salazar-Gomez[1], David Sierra-Gonzalez[1], Manuel Diaz-Zapata[1,2], Anshul Paigwar[1], Wenqian Liu[1], Ozgur Erkent[1,3] and Christian Laugier[1]

***Abstract*— Semantic grids are a succinct and convenient approach to represent the environment for mobile robotics and autonomous driving applications. While the use of Lidar sensors is now generalized in robotics, most semantic grid prediction approaches in the literature focus only on RGB data. In this paper, we present an approach for semantic grid prediction that uses a transformer architecture to fuse Lidar sensor data with RGB images from multiple cameras. Our proposed method, TransFuseGrid, first transforms both input streams into top-view embeddings, and then fuses these embeddings at multiple scales with Transformers. Finally, a decoder transforms the fused, top-view feature map into a semantic grid of the vehicle's environment. We evaluate the performance of our approach on the nuScenes dataset for the vehicle, drivable area, lane divider and walkway segmentation tasks. The results show that TransFuseGrid achieves superior performance than competing RGB-only and Lidar-only methods. Additionally, the Transformer feature fusion leads to a significative improvement over naive RGB-Lidar concatenation. In particular, for the segmentation of vehicles, our model outperforms state-of-the-art RGB-only and Lidar-only methods by 24% and 53%, respectively.**

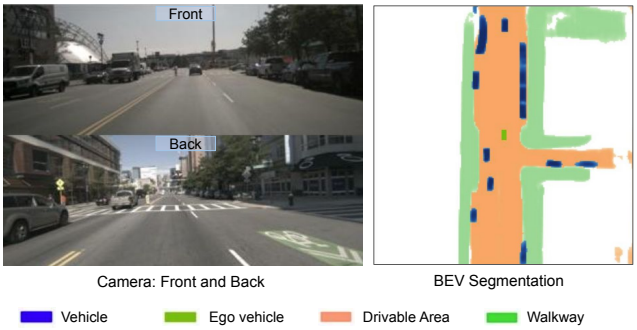


Fig. 1: We propose an architecture to fuse Lidar and multi-camera data for semantic grid prediction. The image on the left shows the front and back cameras from a validation sample of the nuScenes dataset. The image on the right shows the semantic grid predicted by our model.

I. INTRODUCTION

Semantic grids are Birds-Eye-View (BEV) representations of the world where each cell of the grid has an associated class (e.g. road, vehicle, crosswalk) probability distribution. They constitute a compact and simple way of representing a complex 3D environment. A simplified version of semantic grids, where only the occupancy for each cell is predicted, has been used for a long time in the mobile robotics domain due to its simplicity and lightweight computational requirements [1]. With the increase in computational power and the introduction of large-scale robotics datasets [2], [3], researchers have turned their attention to the prediction of semantic grids, which enable improved situation understanding [4], [5], [6], [7].

We can classify the semantic grid prediction methods based on the modality of their input data; some works rely exclusively on camera data [7], [5], [8], [9], [10], [11], others on Lidar range data [6], and finally some approaches fuse multiple sensor modalities to predict the grids [4], [12]. Some other methods explore Lidar-camera fusion for 3D object detection, either by relying on the sensor transformation [13], [14] or, more recently, using cross-attention [15], [16].

Camera-only approaches differ mainly only on how they find the correspondence between image pixels and cell locations in the grid. For example, the authors from Lift-Splat-Shoot (LSS) propose a depth prediction task for each pixel in the image [5]. Other options include the Orthographic Feature Transform (OFT), that projects all voxels above a given cell onto the image and averages the corresponding features [17], [7]; the View Parsing Network (VPN) which relies on fully connected layers to learn the transformation [10]; and Cross-view Transformers that use dot product attention with learnable top-view queries to find the corresponding image features [11].

In contrast to camera-only approaches, Lidar-only methods can easily identify the input data associated to each cell [6]. However, point clouds present difficulties at large distances due to sparsity, something that does not affect high-resolution images. Moreover, certain classes such as sidewalk, lane markings or grass are intuitively easier to identify from camera data than from range data.

To leverage the advantages of both modalities, certain approaches propose fusion architectures [4], [12], [15]. This idea follows the trend observed in the 3D object detection field, where the dominance of fusion approaches over single modality methods is clear by observing online leaderboards [3], [2]. To approach Lidar-camera fusion for semantic grids, Erkent et al propose to first transform the point cloud onto an occupancy grid, and then fuse this representation with a set of semantic grids obtained by projecting the semantically-segmented image onto ground planes at different heights [4]. Hendy et al proposed to fuse the encoded features from

This work was partially supported by Toyota Motor Europe.

[1]The authors are with Inria, Univ. Grenoble-Alpes, Grenoble, France {firstname.lastname}@inria.fr

[2] Manuel Diaz-Zapata is with CITILab, INSA Lyon.

[3] Ozgur Erkent is with Hacettepe University.

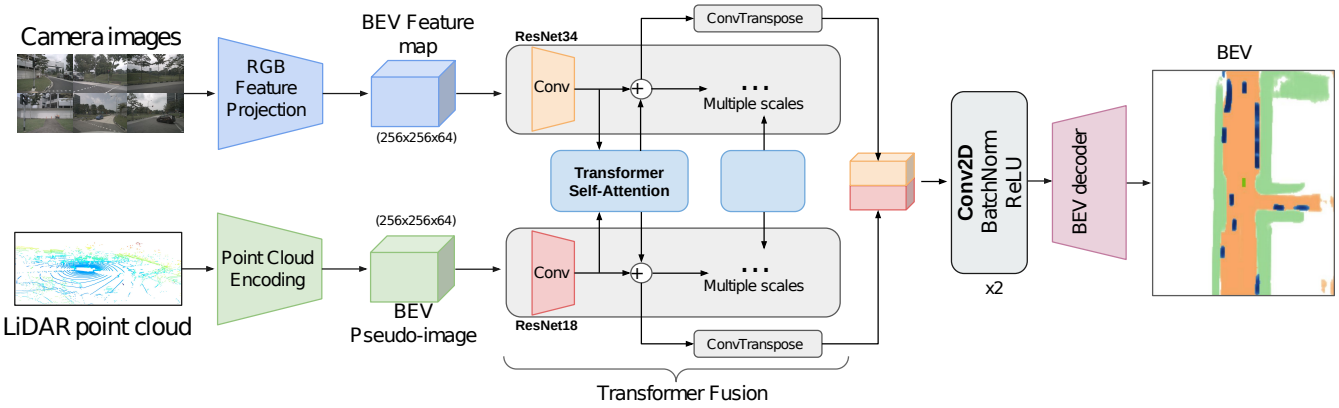


Fig. 2: Proposed architecture for semantic grid prediction. First, features are extracted from the multiple RGB images using a RGB Feature Projector architecture [5], and from the Lidar data using a Point Cloud encoding structure [18]. Both feature streams are then processed by a transformer based multi-scale architecture [19]. At different scales, a transformer is used to fuse Lidar and camera features and its output is aggregated to the unprocessed features and up-scaled with a ConvTranspose layer. All intermediate feature maps are then concatenated, processed by a Conv2D layer, and finally used to generate the bird's-eye-view with the BEV decoder.

multiple cameras using a VPN to generate the semantic grid [12]. Then, this grid is fused using heuristics with another semantic grid obtained from Lidar data. More recently, Li et al propose to fuse camera and lidar mid-level features using the cross-attention mechanism to detect 3D objects [15].

In this work, we further explore the problem of semantic grid prediction using fused Lidar-camera data. In particular, we propose to leverage a transformer-based architecture for this task [20], [19]. The ability of transformers to fuse multi-modal sensor inputs has been recently corroborated with the Transfusion architecture, which obtained state-of-the-art performance for the 3D object detection task in the nuScenes dataset [16]. An earlier approach, the TransFuser, proposed an end-to-end planning architecture that took single camera and Lidar data projected on the ground and fused it at different scales using transformers [19]. Inspired by this work, in this paper we propose to fuse multi-camera and Lidar data for the semantic grid prediction task.

We can outline our contributions as:

- We propose a transformer-based, multi-scale fusion architecture to fuse multi-camera and Lidar features and predict semantic grids.
- The proposed method outperforms state-of-the-art Lidar-only and camera-only competing approaches.
- The proposed multi-scale fusion mechanism outperforms a baseline naive fusion, where the different modalities are simply concatenated.
- We thoroughly analyze how the hyperparameters of the model (e.g. the number of fusion scales) impact the performance of the model.

II. METHOD

In this section, we present TransFuseGrid, our sensor fusion approach for the semantic grid prediction task. We start by formalizing the problem. We are given a set of n monocular images $\{I_k\}^{k=1\dots n}$; a set of Lidar returns $\{\mathbf{x}_1, \dots, \mathbf{x}_P\}$, where P is the non-constant number of returns and $\mathbf{x}_i \in \mathbb{R}^4$ represents the 3D location of each return and its intensity; and the corresponding transformation matrices between the sensors. The goal is to predict a top-view segmentation of the scene; in this paper we predict individual semantic grids for each class $G_C^{W\times H}$, where each $g^{(i,j)}$ represents the probability of that cell belonging to the class, and W, H represent the width and height of the output grid, respectively.

The proposed approach predicts semantic grids by fusing the Lidar and multi-camera input. The data from these two sensor modalities contains complementary information but is inherently different in terms of structure, perspective and density. Hence, fusing both modalities optimally for the downstream task constitutes a challenge. To overcome this, we propose a three-stage network design, as illustrated in Fig. 2. In the first stage, we convert and align these two data modalities into a common representation. The RGB Feature Projector is used to process camera images and project features onto the BEV plane [5]. Similarly, the Point Cloud encoding network processes the 3D point cloud to generate a BEV pseudo image [18]. In the second stage, these two feature representations are fused at multiple scales using a transformer based multi-scale network, technique seen in [19]. Finally, in the last stage, fused intermediate features are up-scaled with a transpose convolution and passed through a BEV Decoder to generate the semantic grids.

A. RGB Feature Projection

Given an arbitrary number of cameras, to infer the transformation from their respective image plane to a common BEV representation, we use the recently proposed state-of-the-art model Lift-Splat-Shoot [5]. This architecture "lifts" each image into a frustum of features for each camera and then "splats" all frustums into a bird's-eye-view grid.

TABLE I: Results on the NuScenes validation split. We compare the Intersection over Union (**IOU**) of the generated semantic grids with the competing approaches in validation set. Best results are presented in **bold** font, second best in Blue. (*: LSS model trained by us, using the source code given by the authors.)

Models	Vehicle	Drivable area	Lane divider	Walkway
LSS (pre-trained) [5]	32.80	-	-	-
LSS * [5]	28.94	61.98	**37.41**	50.07
Pillar feature Net [18]	23.43	69.19	26.05	30.57
TFGrid (concat)	32.88	74.18	30.41	43.78
TFGrid	**35.88**	**78.87**	35.70	**50.98**

TABLE II: Results on the NuScenes validation split. We compare the IOU of the competing approaches in only **Night** conditions and only **Rain** conditions. Best results are presented in **bold** font, second best in Blue.

Class	Vehicle		Drivable area		Lane divider		Walkway	
Model	Night	Rain	Night	Rain	Night	Rain	Night	Rain
LSS (pre-trained) [5]	31.06	34.06	-	-	-	-	-	-
LSS * [5]	28.41	30.81	49.13	56.86	**15.87**	**34.72**	24.56	**46.59**
Pillar feature Net [18]	28.54	19.56	63.89	64.01	3.60	23.72	23.52	27.22
TFGrid (concat)	37.89	33.52	68.98	67.25	7.27	27.90	23.60	40.98
TFGrid	**39.33**	**37.14**	**70.45**	**69.74**	15.17	30.25	**24.65**	44.26

In our approach, as in the original work, the input images are resized and cropped to a shape of 128×352 ($H \times W$). The size of the output BEV feature map is set to a resolution of 256×256, with a depth of $C = 64$ features. This is to maintain the same shape through the different stages in the architecture, as our final semantic grid is of size $128m \times 128m$ with a resolution of $0.5m$.

B. Point Cloud Encoding

To encode a 3D scene into a BEV feature map we use a Pillar feature encoder [18]. This model discretizes the point cloud into an evenly spaced grid in the BEV plane, creating a set of bins or pillars on which the points are clustered. The points in each pillar are augmented by distance to the arithmetic mean and offset from pillar center operations, for total dimensions of 9 for each point. Then, using a simplified version of PointNet [21] on the set of points within each pillar, a BEV pseudo-image with the encoded features is created.

In our proposed model, we set the maximum number of pillars to 10000, and the maximum number of points per pillar to 100. The point cloud range has shape $128m \times 128m$, setting pillars in x and y from $-64m$ to $64m$, with a resolution of $0.5m$. The result is a BEV feature map of shape $256 \times 256 \times C$, with $C = 64$ being the number of features per pillar.

C. Multi-scale fusion with Transformers

To fuse BEV feature maps obtained from camera and Lidar data we use a self-attention transformer at different scales. The core idea is to exploit the self-attention mechanism to learn the optimal fused features from the image and Lidar streams. Compared to CNN-based architectures, transformers have shown an increased capacity to learn global representations that retain the long-term dependencies between the input features. This is convenient for the Lidar-camera fusion task, as the flattened Lidar and multi-camera feature maps have typically a substantial size.

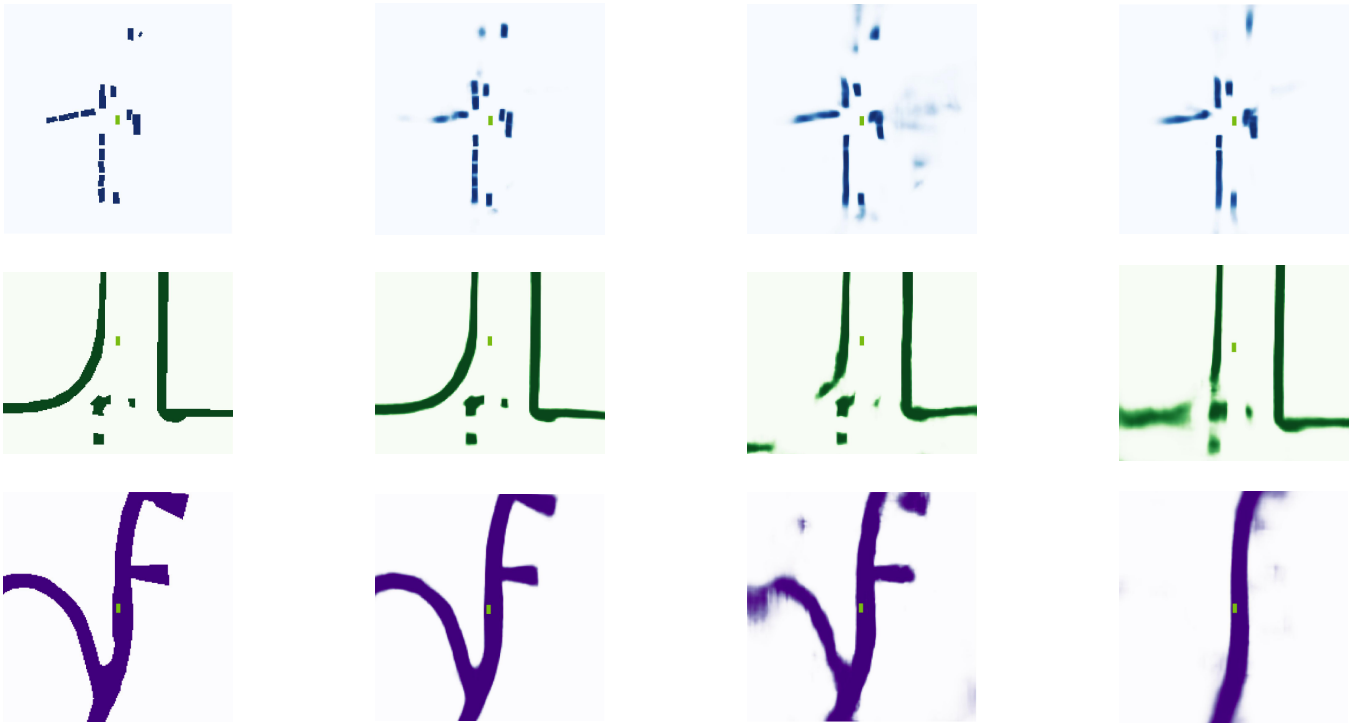
We use ResNet-18 and ResNet-34 as convolutional feature extractors [22], where spatial biases of the scene are encoded at multiple scales. These features are sent to the transformer module, and then fed back to the branch using an element-wise summation with the current features.

In our approach, the convolutional feature extractors receive as input the feature maps produced by the RGB Feature Projection and Point Cloud encoders. These maps are individually processed by different ResNet networks, and their intermediate features are fused by a transformer module. After the element-wise summation at each scale, the features are up-scaled with a transpose convolution layer, which outputs a feature map of size $256 \times 256 \times C$, where C represents the features depth at each stage $(64, 128)$, as shown in Fig. 2.

D. Semantic Grid Generation

The intermediate features obtained at different scales from the Transfuser model are converted to a same size feature map of 256×256 using transposed convolutions. These feature maps are concatenated resulting into a final feature map with 384 features. A sequence of two convolutional blocks (Conv2D, BatchNorm and ReLU), are then applied to obtain the final input for the BEV decoder.

The BEV decoder consists in a combination of ResNet blocks, using initially a Conv layer with stride 2 and kernel size 7, followed by BatchNorm and ReLU. Finally, two up-sampling operations restore the 256×256 resolution and, after two more convolutions, the final semantic grid is generated.



(a) Ground truth (b) TFGrid (ours) (c) TFGrid (concat) (d) LSS*

Fig. 3: Comparison between ground truth and semantic grid predictions for vehicles (top row), walkway (second row) and drivable area (third row) for different samples in the validation split. We display the class probabilities as a colorscale, without thresholding. In the top row, we can see how the models with Lidar input can detect all vehicles in the scene, while LSS fails to do so, probably due to occlusions. Furthermore, the definition of the detected vehicles with the TFGrid model is much cleaner than TFGrid concat model. In the second row, we see how TFGrid out performs both baselines for walkway class. In the third row, the ability of the models with Lidar input to predict the details of the drivable area is clearly superior to that of LSS. The bottom row shows the combination of the three classes and results compared with the ground truth

E. Training

For each semantic class, we train a model using Binary Cross Entropy as loss function. As optimizer we use Adam with a learning rate value $1e-3$ and weight decay $1e-7$ [23], with batch size 2 all models are trained for 800k steps. We use PyTorch framework for coding the model. The source code will be open-sourced upon publication at github [1].

III. Experiments

In this section, we illustrate the experiments performed with the proposed model. To evaluate the performance of the proposed model we compare against competing and baseline models in general settings and in challenging scenarios such as night and rain.

[1]https://github.com/gsg213/TFGrid

A. Experimental setup

Dataset: We use NuScenes for training and validating our approach [3]. NuScenes is a large dataset containing 1000 scenes of 20 seconds each, with a variety of different sensors, including cameras and Lidar. This dataset has a diversity of classes, locations, day and night time driving and weather conditions. For training and evaluation, we focus on the `drivable_area`, `walkway`, `lane_divider` and `vehicle` classes only, as these are some of the most crucial from the autonomous navigation perspective.

From the validation data, we extract scenes with challenging conditions such as night-time driving and rain to further evaluate the performance of our model.

Metric: We use Intersection Over Union (IOU) as a metric to evaluate the performance of our model. This metric allows us

TABLE III: IOU results for different number of transformer fusion modules.

Model	Transformers	Vehicle	night	rain	Drivable area	night	rain	Time (ms)
TFGrid	1T	34.48	**40.34**	35.62	76.68	68.41	69.72	47.90
TFGrid	**2T**	**35.88**	39.33	37.14	**78.77**	**70.45**	**69.74**	54.36
TFGrid	3T	34.38	37.56	34.39	72.55	64.67	63.79	61.44
TFGrid	4T	35.86	40.11	**38.71**	76.13	68.16	67.67	66.21

to quantify the amount of overlap between the ground truth mask generated from the nuScenes dataset and the predicted semantic grids.

Baselines and ablation study: We compare our approach TransFuseGrid (**TFGrid**) to diverse competing and baseline models. To show that the fusion of sensor modalities improves over single-modality variants, we train two different models. The first one is **Lift-Splat** [5], which uses only camera data. The second one is **Pillar Feature Net** [18], which uses only point cloud data. In both of these models, the Transfuser module is removed and the output feature maps are directly passed through the BEV decoder to predict semantic grids.

To evaluate the advantage achieved by using transformers in the data fusion process, we train another model **TFGrid (concat)**, where the RGB Feature Projector features are concatenated with the Point Cloud encoding features and directly passed through the BEV decoder.

The results presented in Tables I, II and Fig. 3, represent the results for TransFuseGrid with 2 transformers.

We present ablation studies on the number of scales and transformers used during the BEV feature map fusion. This way, we are able to understand its role in the fusion at multiple scales in the architecture, varying from using just 1 transformer to 4. Each scale represents a different resolution in the intermediate features, where 1 transformer uses only the first Conv block from ResNet with a feature resolution of 64×64, and 2 transformers use the first two Conv blocks from each ResNet, with feature resolutions of 64×64 and 32×32, respectively. Similarly for 3 and 4 transformers.

Finally, a last test is performed in the ablation study. The BEV pseudo-image from the Point Cloud Encoding is concatenated to the features after the Transformer fusion step, aiming to enhance the performance of the model by adding uncorrupted features just before the bird's-eye-view generation.

B. Results

Table I shows the quantitative results obtained by our model in the nuScenes validation split. For the vehicle class, we can observe how our model (TFGrid) outperforms all competing approaches in general conditions. The larger improvement over camera-only models is at night, and over Lidar-only and concat fusion models in rain conditions, as seen in Table II. This shows how our model learned to leverage the multi-modal data better than with the naive concatenated fusion.

In Table I, we can also see how the proposed model obtains better results in all conditions for the drivable area class, followed by the TFGrid concatenated fusion model. We can observe how the models that combine both modalities, outperform the camera-only and lidar-only methods.

In the third column, we can observe the results obtained for the class Lane divider. For this class, the camera-only method obtains better results than our model, which ranks second and above the baselines. We hypothesize that for this class, the Lidar features are introducing noise from the different obstacles (e.g. vehicles) that often occlude the lines.

In the last column of Table I, the results show that the proposed model performs better for the walkway class, followed very close by LSS.

Table II shows the comparison for challenging scenarios, such as night and rain conditions. TFGrid outperforms the baselines in all categories except lane divider, which is in line with the results for the same class in the complete validation set.

In Fig. 3, we present in the first row a comparison for vehicle segmentation between our model and the baselines. The qualitative results show that LSS is not able to cleanly segment vehicles located far away, as well as those that are occluded. The semantic grid generated by our approach shows superior results, finding occluded vehicles, and demonstrates better defined and positioned vehicles in the scene, whether they are close or far-away. This clearly highlights how the model leverages the depth information from the point cloud thorough the fusion operations.

The second row presents the semantic grid for the walkway class. The results evidence the superior performance of the proposed model over the baselines. Furthermore, we can observe how for the camera-only method some segments are not shown in the generated grid.

In the third row, we show the drivable area representation produced by each model. These qualitative results show a superior performance by the models that fuse Lidar and camera data. We additionally provide an anonymous public video submission illustrating the results in different scenarios for classes vehicle and drivable area, comparing TFGrid and the baselines [2].

In Table III we present the results from the ablation studies at different scales and with different number of transformers. We can observe that, in average, the best model for the classes vehicle and drivable area is the one using only 2 scales at resolutions of 64×64 and 32×32.

A last test is performed in the ablation study. The features from the Point Cloud Encoding are concatenated to the features after the Transformer fusion step, the purpose of this

[2]https://youtu.be/j2nBqtwcCOo

change in the architecture is to improve the performance of the best model. The model with 2 transformers is modified, but the results show that concatenating these features has not an improving effect, by contrast, it has a negative effect in the semantic grid prediction.

The model using only 1 scale presented the lower inference time at 47.90 ms, which translates into 20.87 Frames per second (FPS). As the model with 4 scales requires more intermediate features and has more convolution blocks in the transformer fusion step, it presents the highest inference time at 66.21 ms, which translates into 15.10 FPS. TFGrid requires a voxelization of the point cloud, which takes a time of 5.34 ms. These times were obtained on a single AMD Radeon Instinct MI50 GPU.

IV. Conclusions

In this work, we have presented a transformer based multi-scale model to fuse Lidar-RGB data for semantic grid prediction. Based on our experimental evaluation, our model achieves superior performance than RGB-only and Lidar-only models for vehicle, drivable area and walkway classes on all weather/lighting conditions. . In particular, for the sementation of vehicles, our model outperforms the competing camera-only and Lidar-only models by 24% and 53%, respectively.

Furthermore, compared to a naive concatenated fusion model, the proposed approach produces significantly better results on all classes, highlighting the performance boost introduced by the multi-scale fusion approach.

On the lane divider class, our model lags slightly behind the camera-only method. We hypothesize that for this class, the Lidar features introduce noise due to the different obstacles that overlap the lines. Future work will examine additional classes and network configurations to confirm our hypothesis.

ACKNOWLEDGMENT

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 (see https://www.grid5000.fr).

References

[1] A. Petrovskaya, M. Perrollaz, L. Oliveira, L. Spinello, R. Triebel, A. Makris, J.-D. Yoder, N. Urbano, C. Laugier, and P. Bessière, "Awareness of Road Scene Participants for Autonomous Driving," in *Handbook of Intelligent Vehicles*, A. Eskandarian, Ed. Springer, 2012, pp. 1383–1432. [Online]. Available: https://hal.inria.fr/hal-00683761

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[3] 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.

[4] Erkent, C. Wolf, C. Laugier, D. S. Gonzalez, and V. R. Cano, "Semantic grid estimation with a hybrid bayesian and deep neural network approach," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 888–895.

[5] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Proceedings of the European Conference on Computer Vision*, 2020.

[6] J. Fei, K. Peng, P. Heidenreich, F. Bieder, and C. Stiller, "Pillarsegnet: Pillar-based semantic grid map estimation using sparse lidar data," in *IEEE Intelligent Vehicles Symposium (IV): 11-17 July 2021, online*. Institute of Electrical and Electronics Engineers (IEEE), 2021, p. 838–844.

[7] 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 (CVPR)*, June 2020.

[8] 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.

[9] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 445–452, 2019.

[10] 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.

[11] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 760–13 769.

[12] 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.

[13] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 663–678.

[14] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] 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.

[16] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 1090–1099.

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

[18] 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.

[19] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7073–7083.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.