Semantic and Elevation BEV Predictions for Robotic Perception

Michael Lötscher and Ryan Slocum

***Abstract*— Mobile robots require a holistic understanding of the semantics and geometry of their surroundings for tasks such as route planning, navigation, and traversability estimation. Bird's-Eye-View (BEV) maps provide a particularly useful spatial representation for such tasks, offering a unified top-down view. However, the learned estimation of BEV for ground robotic platforms has not been thoroughly explored. In this work, we adapt an existing state-of-the-art (SOTA) model [1] to a recently developed mobile robotic perception dataset [2] to estimate a Bird's Eye View (BEV) representation of the semantics and elevation of the environment. We do so with the goal of creating a useful benchmark for future researchers using this dataset.**

I. INTRODUCTION

Bird's-Eye-View (BEV) projection is a useful technique in robotic perception, involving fusing multi-sensor data into a BEV map of the environment. This modality is particularly useful for mobile robotics, due to its applicability for path planning, exploration, and obstacle avoidance tasks. Because of the simplified top-down view and the unified representation of multi sensor information, BEV maps can make these tasks simpler and more tractable, as opposed to doing path planning through a 3D voxel occupancy grid. Additionally, BEV maps are a useful artifact for human understanding of a scene, as humans are accustomed to navigating via BEV maps.

Although there is a substantial body of work in visual BEV projection for training and evaluation on autonomous vehicle (AV) datasets [3]–[7], mobile ground robots (such as legged and wheeled platforms) have not yet seen broad application of these BEV methods. This is in large part due to the lack of large-scale datasets for mobile robotics. Research for AV perception has benefited from projects like the NuScenes [8] and Waymo [9] datasets, but so far there has been no such dataset for mobile robotics. To this end, researchers at the ETH Zürich Robotic Systems Lab have developed the TartanGround dataset, with more than 1.4 million samples [2].

This project aims to extend an existing SOTA architecture such as PointBeV [1], which was built to perform BEV occupancy prediction for autonomous vehicles, and train it to perform semantic and elevation mapping tasks on the TartanGround dataset. These tasks, in particular semantic mapping, are critical for robotic mobility in an environment because they allow the robot to estimate the traversability cost of paths it plans to take. Because the very recent release of the TartanGround dataset, there is little on learned BEV with synthetic data for mobile robotics. With these adaptations, we hope to set a benchmark for future researchers to improve on.

The contributions of this work are as follows.

- A pipeline that processes synthetic depth, semantic and RGB images to produce BEV maps labels for training and validation.
- An extension of an existing architecture to predict multiclass semantics and 2.5D elevation maps on a synthetic dataset.
- Training and evaluation of three different models, one for each different type of environment in the dataset: industrial, urban, and natural.

II. RELATED WORK

A. BEV projection

In existing BEV projection works, there are two main methods for projecting camera features into the BEV space. The first, pioneered by Lift Splat Shoot [6], *lifts* every feature along its respective camera ray using a predicted depth distribution, and *splats* those contributions into the corresponding BEV grid cells. This approach influenced many other papers, including [3], [5], [10]–[12].Although these methods do outperform separate monocular depth estimation compared to BEV projection [6], the process of lifting and splatting every feature is slow and expensive.

More recently, an approach for feature pulling has been developed. In this method, points in the BEV plane are formed into vertical columns, and the voxels that form the columns are then used to form queries into the image plane, after which the queried features are aggregated back into the BEV cell. This method is used in [4], [13], [14]. These feature-pulling methods frequently employ transformers, leading to better results but also longer training times due to the increased model complexity.

In this work, we are choosing to base our project on PointBeV [1], a recently published architecture that uses the feature-pulling approach to BEV projection. A major advantage of PointBeV is its implementation of a sparse view transformation leveraging sparse point sampling, which greatly reduces the memory footprint and training time of the model while maintaining performance consistent with state-of-the-art models in binary semantic segmentation tasks [1]. The authors' source code is open source and available for adaptation. For these reasons, we choose to construct our baseline with this model as the starting point. However, PointBeV does not support multiclass semantic segmentation and 2.5D elevation mapping, so in this work, we extend it to support these tasks and train the model on the TartanGround dataset.

B. Datasets

Although there exist many large datasets with multimodal data and quality annotations for autonomous vehicles (well known ones include NuScenes [8] and Waymo [9]), there exist far fewer for ground mobile robotics. Those that do exist are relatively small, with less than 10,000 frames [15], [16], and not well-suited for training deep-learning models.

The comparatively large TartanGround [2] dataset offers more than 1.4 million frames, making it much more useful for deep learning research. However, the synthetic data does come with its own challenges, such as inconsistent semantic labels and gaps from simulation training to deployment on a real robotic platform. Because it is a new dataset, we aim to set a baseline for future researchers to improve upon for BEV tasks.

III. METHODOLOGY

A. BEV preprocessing

Because BEV maps are not included in the TartanGround dataset, we needed to generate training data from the sensor information available in the dataset. We have access to the RGB, depth, and labeled semantic images, as well as robot poses at each sample in the trajectory. To generate the necessary BEV maps, we take the following steps.

1) Cluster semantics classes based on navigability: Given the diversity of semantic classes in the dataset and the intended use of the BEV maps we are generating (robotic mobility), we cluster the classes based on their navigability. Some of the resulting classes are *navigable_flat* and *ambiguous_vegetation*, representing the estimated traversability of the class and the larger category it falls into. These clusters are made once for all the data.

2) Construct 3D point cloud from depth and image data: We then use the utilities from the TartanGround package to construct a point cloud of the environment from all the viewpoints we have seen in our trajectory. Each point is labeled with the relevant semantic class.

3) Crop and rotate the point cloud per robot pose: For each pose in the trajectory, we then crop the point cloud to the dimensions of our desired map. For the model we have trained, we use a 20x20 meter grid and consider only points less than 3 meters above or below the robot's camera frame.

4) Compute ground and ceiling layers: We extract a maximum ground layer as in [10]. For each grid cell in the map, we take the lowest point as the minimum ground layer. We then ascend the column until there is a gap of more than 1.5 meters between points, at which point we take the lower point before the gap as the maximum ground layer and the upper point after the gap as the ceiling layer. In this work, we only consider the maximum ground layer, as it determines where a ground robot could reasonably move, but future works could also utilize the minimum ground and ceiling layers.

5) De-noise and post process semantic and elevation maps: Using the semantic and elevation maps, we apply a smoothing kernel to the semantics in order to reduce noise.

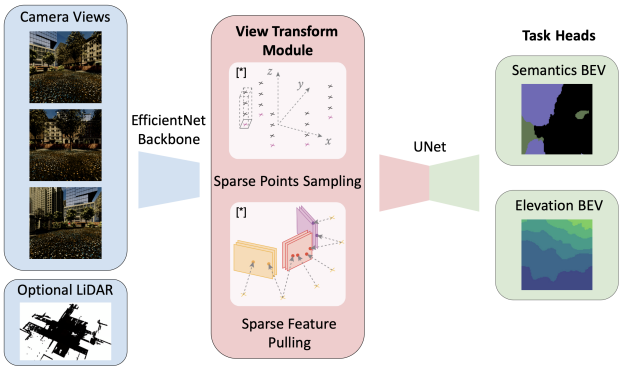

Fig. 1. PointBeV model architecture, with the proposed modified task heads. Center images with * from [1]

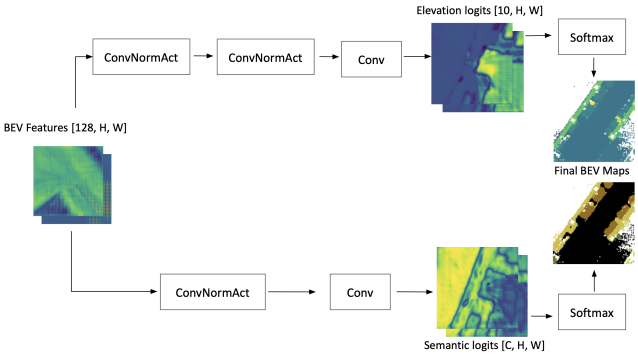

Fig. 2. The modified task heads presented in this work

This is especially important for natural environments, where there are many small features, such as low lying grasses, that add noise to the maps and uncertainty when training the model.

B. Camera setup

Although the TartanGround dataset includes tools to easily resample the RGB images to different camera intrinsics and extrinsics, we chose to retain the original four surround cameras that were available in the dataset, to make it simpler for future researchers to replicate our results. These four cameras were oriented front (0°), left (90°), back (180°), and right (270°), each with a 90° field of view (FOV) and a pinhole camera model. There is no overlap between the camera FOVs.

C. Model Architecture

As mentioned above, this work builds on the architecture presented in PointBeV [1]. In Figure 1, this architecture can be observed with the modified task heads presented in this work.

The inputs to the network are primarily the camera images and camera calibration parameters. LiDAR is also an optional input, fed to the model in the same BEV representation as the output map, with LiDAR hits collapsed into a plane.

The RGB images are fed into a pretrained EfficientNet backbone, which extracts camera features. The feature-pulling method described above is used at this point. The

point sampler selects BEV grid cells, forms columns of voxels in those grid cells, and queries the related camera features in the feature-pulling step. These features are then collapsed into a 2D BEV view. The optional LiDAR data is used to improve the efficiency of the sampling step. During training and inference, we follow PointBeV's two stage *coarse/fine* sampling strategy. The coarse pass uniformly samples a small subset of BEV grid cells, whereas the following fine pass densifies around the highest confidence anchor points found in the coarse pass.

Once we have these features in the BEV view representation, we feed them into a UNet that encodes and decodes the map into a set of unified BEV features.

We then have two different task heads that take these unified BEV features as input. Both heads, illustrated in Figure 2, consist of concatenated convolutional layers, ending with a softmax and the extraction of the most likely class or elevation bin.

D. Training

To test the effectiveness of our pipeline, we trained three different models, each in a different type of environments in the TartanGround dataset. Figure 3 shows a visual overview of a selection of the environments used. The motivation behind this split was that each type of environment could have different semantic groupings and that it would be easier for each model to learn how the associated semantics. In addition, it is interesting to compare the results from more structured environments (industrial and urban) to those from unstructured environments (natural).

The models were trained using the Adam optimizer, with an adaptive learning rate and using cross-entropy loss. Each model was trained for 100 epochs on 4 NVIDIA RTX4090s. Due to the different sizes of the datasets, the training time took from 6 to 12 hours. We trained each model on 5-8 environments, leaving one exclusively for validation and one exclusively for testing. Because the train, validation, and test splits were made using entire trajectories and environments, there is a considerable amount of variance in the amount of samples in each split for each model.

Among the three models, there was about 180 GB of training data in total, yielding the frames listed in Table I.

Fig. 3. Example selection of environments each model was trained on

TABLE I
NUMBER OF FRAMES IN THE DATA SPLIT

Model	Train	Val	Test	Total
Industrial	6 028	1 616	2 686	10 330
Urban	3 814	3 863	2 653	10 330
Natural	6 735	3 369	3 243	13 347
Total	16 577	8 848	8 582	34 007

IV. RESULTS

Evaluation of the models on their respective test sets reveals interesting findings, which we will explore in the following subsections. Figure 4 illustrates an example of inference on both semantics and elevation tasks. Table II details the metrics that we used to evaluate the models: mean Intersection over Union (mIoU) and frequency-weighted Intersection over Union (FWIoU) for the semantic segmentation task, and Mean Absolute Error (MAE) for the elevation task.

A. Semantic task head

Semantic segmentation proved to be a challenging task for this model. Quantitatively, the mIoU for each of the models is quite low, whereas the FWIoU is comparatively high. Both of these metrics were included because in many cases, the number of pixels containing one of the less frequent classes was rather small, and frequent classes such as *navigable_flat* made up the majority of many images. Qualitatively, the model seems to identify larger swaths of uniform terrain well, but high-frequency details, such as grasses or smaller rocks, are frequently lost.

A challenge encountered during the training of these models was that the semantic classes provided in TartanGround are fairly limiting from the point of view of determining traversability for ground robots. For example, in the *ForestEnv* environment, both large obstacles such as trees and small, possibly navigable objects such as low grass were labeled as *plant*. So, despite the fact that the end use of this model would likely want a distinction between these types of obstacle, we needed to group them both under *nonnavigable_static*. These distinctions made it difficult to train a model across different environments.

B. Elevation task head

Despite having no access to geometric information, only RGB camera images, the elevation task head performs surprisingly well, with a highest MAE of 0.5 meters in unstructured natural environments.

Qualitatively, the elevation model also struggles with high-frequency details and elements of the environment that are further away from the ego position, but overall performs fairly well on the more structured environments in the industrial and urban models.

TABLE II
RESULTS OF INFERENCE ON TEST SET

Model	Distribution	# Test Frames	Semantic mIoU ↑	Semantic FWIoU ↑	Elevation MAE (m) ↓
Industrial	In train	1 349	**0.328**	0.573	0.230
	Not in train	1 849	0.174	0.682	0.326
Urban	In train	733	0.271	**0.807**	**0.178**
	Not in train	1 920	0.109	0.310	0.401
Natural	In train	596	0.300	0.699	0.240
	Not in train	2 090	0.164	0.416	0.521

C. Lack of generalization

In this discussion, we distinguish between the frames *in-distribution* and *not in-distribution*, which are frames from the test set where the environment was seen before in the training process or not, respectively. This does not mean that the model was trained on images from the test set. Rather, it distinguishes between between environments that were reserved completely for test-time inference. As an example, *ForestEnv* was an environment that had trajectories in both the training and test sets, whereas *GreatMarsh* is an environment that was entirely reserved for validation and testing. This was done to test how the model generalizes to new environments that have not been previously seen.

As seen in Table II, there is a definite difference in the performance of the models on environments that were in-distribution and those that were not in-distribution, with the MAE being almost double in the latter case, and the mIoU being about half.

We have two hypotheses for why this could be. The first is that the model has difficulty transferring its learnings about the clustered semantic classes from one environment to the other. This would be expected given the amount of data we are training on, which is on the order of only thousands of samples (see Table I). This issue could certainly be solved by training on more and more varied data, which is an option for future researchers given the diversity and size of the TartanGround dataset. However, due to limited time and resources, we trained these models with a subset of all available data.

The second hypothesis for why the model is failing to generalize is that it could be learning the geometry of the environment from the training set. While no images are identical between the test and train sets, due to the spatially small area that all the trajectories that are pulled from, it is very likely that the model has seen the environment from similar viewpoints in the test set as it had trained on previously. However, we cannot test whether this is the case without dramatically reducing the amount of data we train on, because the trajectories per environment in TartanGround are overlapping and hence not easily separable so that two different sets see different areas. This is a limitation of the dataset that we unfortunately did not have time to work around.

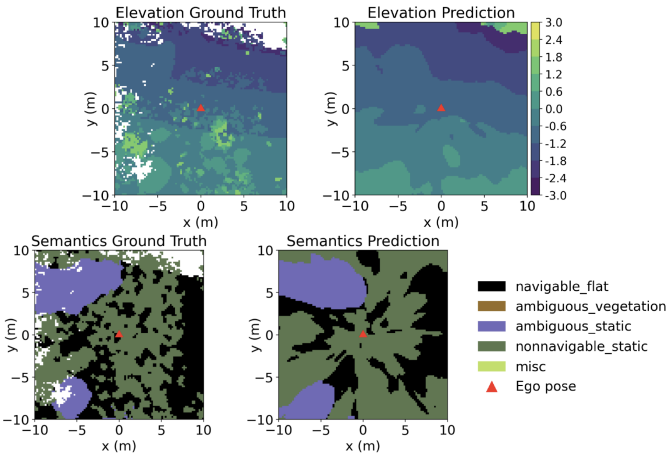

Fig. 4. Representative ground truth vs predictions on the natural model, on the in-training-distribution environment *ForestEnv*

V. CONCLUSIONS

In this paper, we have presented a baseline for future researchers to build upon for BEV mapping tasks in the TartanGround dataset. We trained three different models on a subset of the data, and found that the models do not generalize well to new environments. They can, however, learn quite well the associated semantics and elevation in environments they have already seen. We have also found that there are challenges when using the TartanGround dataset that make it difficult to determine why this lack of generalization is the case.

Future directions for this research could include training data with more precise semantic labeling, reducing spatial overlap between train/test trajectories to better assess generalization, or direct integration of LiDAR data for elevation estimation.

APPENDIX

A. Additional inference examples

See figure 5 for more inference examples.

B. Source Code

The code used to process the data and train the models is available at this GitHub repository:
github.com/leggedrobotics/semantic_bev_mapping

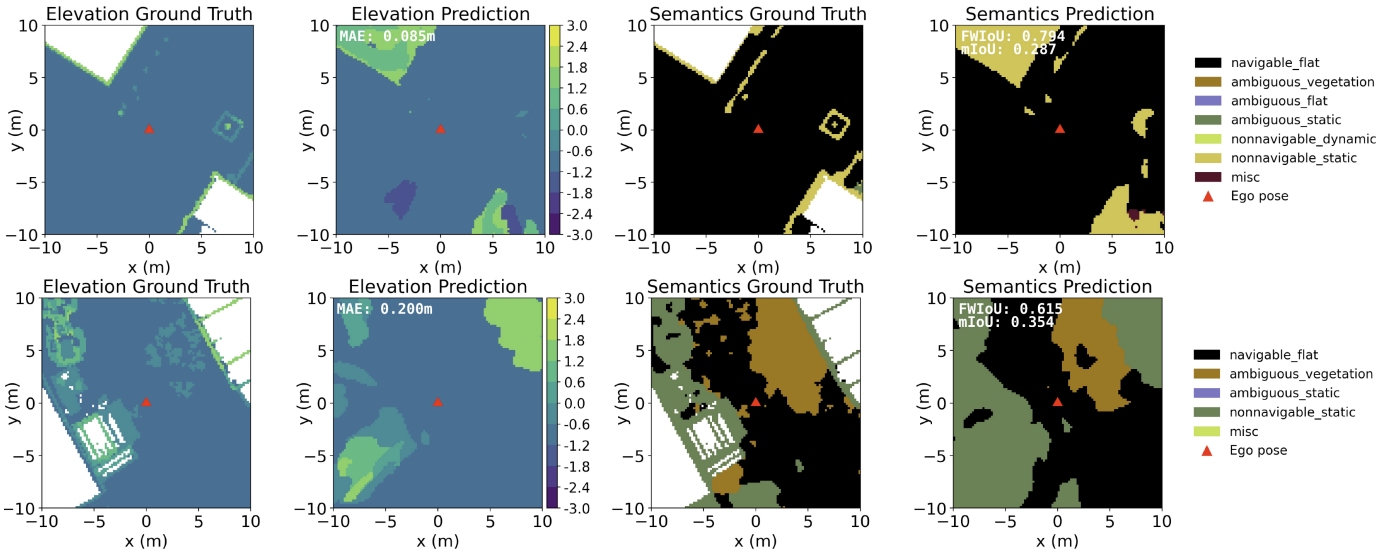

Fig. 5. Two more examples of inference on the test set, from the *OldTownWinter* and *ConstructionSiteDay* environments, which were tested with the Urban and Industrial models.

C. Demonstration Video

For more examples of the performance of the models, see the following video:
polybox.ethz.ch/index.php/s/MKE488JLfbAgggX

References

[1] L. Chambon, E. Zablocki, M. Chen, F. Bartoccioni, P. Perez, and M. Cord, "Pointbev: A sparse approach to bev predictions," 2024. [Online]. Available: https://arxiv.org/abs/2312.00703

[2] M. Patel, F. Yang, Y. Qiu, C. Cadena, S. Scherer, M. Hutter, and W. Wang, "Tartanground: A large-scale dataset for ground robot perception and navigation," 2025. [Online]. Available: https://arxiv.org/abs/2505.10696

[3] J. Frey, M. Patel, D. Atha, J. Nubert, D. Fan, A. Agha, C. Padgett, P. Spieler, M. Hutter, and S. Khattak, "Roadrunner – learning traversability estimation for autonomous off-road driving," 2024. [Online]. Available: https://arxiv.org/abs/2402.19341

[4] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," 2022. [Online]. Available: https://arxiv.org/abs/2203.17270

[5] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," 2024. [Online]. Available: https://arxiv.org/abs/2205.13542

[6] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," 2020. [Online]. Available: https://arxiv.org/abs/2008.05711

[7] J. Zhao, J. Shi, and L. Zhuo, "Bev perception for autonomous driving: State of the art and future perspectives," *Expert Systems with Applications*, vol. 258, p. 125103, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424019705

[8] 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," *arXiv preprint arXiv:1903.11027*, 2019.

[9] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, S. Zhao, S. Cheng, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," 2020. [Online]. Available: https://arxiv.org/abs/1912.04838

[10] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng, G. Okopal, D. Fox, B. Boots, and A. Shaban, "Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation," 2023. [Online]. Available: https://arxiv.org/abs/2303.15771

[11] Q. Jiang and H. Sun, "Lssattn: Towards dense and accurate view transformation for multi-modal 3d object detection," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 6600–6606.

[12] Q. Jiang, H. Sun, and X. Zhang, "Semanticbevfusion: Rethink lidar-camera fusion in unified bird's-eye view representation for 3d object detection," 2022. [Online]. Available: https://arxiv.org/abs/2212.04675

[13] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "Futr3d: A unified sensor fusion framework for 3d detection," 2023. [Online]. Available: https://arxiv.org/abs/2203.10642

[14] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-bev: What really matters for multi-sensor bev perception?" in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2759–2765.

[15] H. Wang, Y. Sun, and M. Liu, "Self-supervised drivable area and road anomaly segmentation using RGB-D data for robotic wheelchairs," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4386–4393, 2019.

[16] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.