BEV Lane Centerlines Using Multi-view Images

Mandeep Singh[1], Sai Ramana Kiran[1], Shiva Kumar Tekumatla[1], Tript Sharma[1]
{msingh2, spinnamaraju, stekumatla, tsharma}@wpi.edu

***Abstract*—Autonomous navigation requires a comprehensive understanding of the traffic scene, including structured representation of the road network and instance-wise identification of the ground plane in the Bird's-Eye View (BEV). In this paper, we propose a method to extract 3D lane centerlines in BEV coordinates from multi-view onboard camera images considering different traffic elements. We take a hybrid approach of integrating classical and modern techniques for positional embedding. The method is particularly useful in heavy traffic situations such as intersections and junctions, where multiple camera views are essential to gather sufficient information.**

Supplementary Material

The supplementary material, code are available at https://github.com/saikrn112/STSU. Original implementation can be found at https://github.com/ybarancan/STSU

I. Introduction

Autonomous vehicles have emerged as a promising solution to improve road safety, reduce traffic congestion, and provide greater mobility for people with disabilities or in remote areas. However, enabling autonomous vehicles to navigate the road network safely and efficiently requires a comprehensive understanding of the traffic scene, including structured representation of the road network [1] and instance-wise identification of the ground plane.

One critical component of scene understanding for autonomous navigation is the extraction of 3D lane centerlines [2] from multi-view onboard camera images. Lane centerlines are used to accurately identify the location and orientation of the road network [3], and they play a vital role in enabling vehicles to safely navigate the road. However, the limited intersecting field of view of onboard cameras, which are typically mounted horizontally to provide a better view of the surroundings, poses a challenge to this task.

Various techniques [6]–[9] have been proposed to address the challenge of lanes, from onboard camera images. These techniques aim to improve the accuracy of the extracted lanes in the bird's-eye view (BEV). However, this task only involves lanes and do not take into account different traffic elements. Moreover, some of the methods only take single front facing camera. Some methods [16] do take all the views to create a topology of different elements but do not consider 3D lanes which are essential in elevated regions. Multi-view images provide different useful information in heavy traffic situations such as intersections and junctions.

Overall, our work contributes to the development of advanced technologies for autonomous driving and has various real-world applications. The effectiveness of our approach demonstrates the potential of combining classical and modern techniques to address challenging problems in this field.

[1] Corresponding authors are from Robotics Engineering, Worcester Polytechnic Institute.

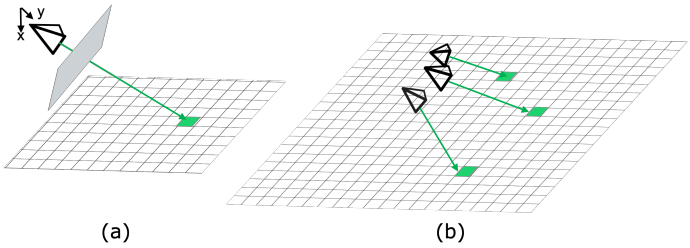

Fig. 1. Positional encoding for different camera views. (a) original approach where for a single camera light rays are projected back into the scene and (b) a New approach which takes extrinsics into consideration for each camera light ray that is projected

A. Contributions

In this paper, we propose a method to extract 3D lane centerlines in BEV coordinates from multi-view onboard camera images. Our approach combines classical and modern techniques to integrate camera views, thereby we attempt to solve the problem posed by the limited field of view of a single front-facing camera. Primarily, we enhance positional embedding for different views by taking the corresponding camera extrinsic parameters with respect to the front-facing camera and improving the learning of traffic structure.

B. Organization of the paper

Rest of the paper is divided as follows, Section II talks more about the inspiration for the problem approaches and the dataset used. Section III gives more insights about the approach to solving multi-view inference and related positional encoding approach. Section IV describes implementation details and outlines ground truth generation. Section V briefly talks about the losses and metrics that we used. Section VI describes results and various insights from them. Finally, section VII ends the discussion outlining challenges and future scope of work for this problem.

II. Related Work

The majority of the existing work can be grouped into the following ways; Lanes estimation, segmentation, scene understanding, and multi-view camera integration. This is an active research problem in the field of self-driving vehicles and many of the works involve classical computer vision modules, modern Convolutional Neural Networks (CNNs), a

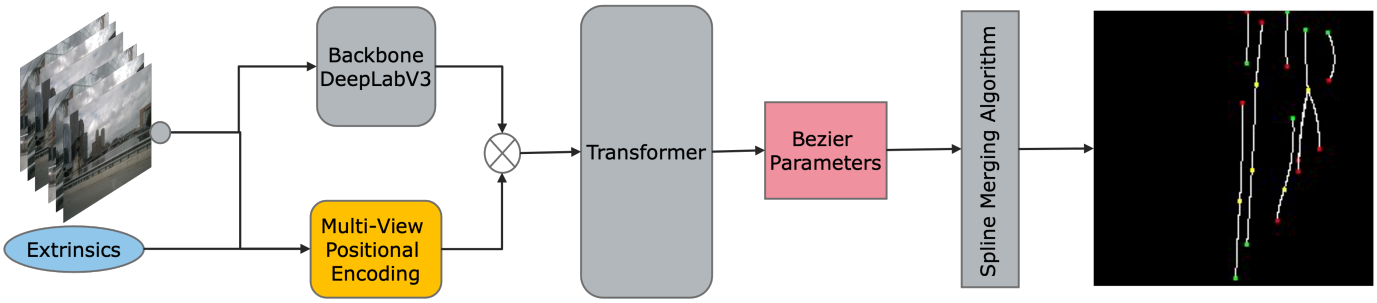

Fig. 2. Modified Transformed Network takes the input of the features from backbone, and multi-view positional encoding. Positional encoding is performed using camera extrinsic parameters. Transformer returns Bezier parameters, which then are used along with spline merging algorithms to return the lane center lines from in BEV

combination of both, and finally powerful and data-hungry transformers.

Detection and Segmentation: The goal of object detection is to predict a set of bounding boxes and category labels for each object of interest. Popularly, YOLO [13] was used as a backbone in many object detection tasks. In recent years DETR [14] quickly rose to prominence because of its robustness. These networks are useful in identifying different traffic elements like cars, objects, pedestrians, traffic lights, etc. which can be incorporated while determining feasible 3D lane center lines.

Lane estimation: Although estimating lanes technically falls under segmentation, identifying lanes is a different problem from a classical standpoint. There is considerable research in this domain using monocular cameras [10]. The task is either performed directly on the image plane [11] or in the BEV plane by projecting the image to the ground plane [12]. [2] is another one of the recent works which aim to predict 3D lanes from a single front-facing camera. It uses a combination of 2D/3D anchor design, which iteratively optimizes till transformers converge.

Scene Understanding: High-definition (HD) map provides abundant and precise environmental information of the driving scene. In [16], authors presented a unified permutation-equivalent modeling approach, i.e., modeling the map element as a point set with a group of equivalent permutations, which accurately describes the shape of the map element, and finally produces robust and stable scene understanding. Similarly, in [15], a directed graph is generated to represent the road network in BEV coordinates. The graph incorporates the road centerlines along with objects such as vehicles, and pedestrians to facilitate a comprehensive understanding of the scene.

Multi-view integration: There are multiple ways explored to merge multi-view images to get a unified scene representation. In [6] n images are fused using the frustum of point clouds generated by extrinsic and intrinsic parameters and converted to BEV. Cross-view transformers(CVT) [8] use Transformer modules to extract unified representation from separate sequences of image features of each view in multi-view integration.

Based on the above literature, there are 3 potential ways to solve the posed problem,

1) Fuse multi view images to get the BEV and then predict the lane centerlines on this BEV. Challenge in this method is to create a network which takes in BEV and outputs the lane centerlines.
2) Directly predict the lane center lines from a single view image and then integrate the predicted lane centerlines from multiple images. However, this approach is fallible as aligning the piecewise centerlines could be difficult and it can miss a lot of lane lines.
3) Instead of staggered approach of predicting one prior to the other, we can fuse and predict the 3D lane center lines directly from one network.

Our method uses 3 approach and is closely aligned with the STSU [15]. Additionally we take [6] methodology of integrating different view images and apply it to the positional encoding of the STSU architecture

A. Dataset

The nuScenes dataset is a public dataset used for autonomous driving research, featuring high-resolution sensor data from a full suite of sensors including lidar, cameras, and radar. It includes over 1,000 scenes with complex urban environments, annotated with 3D object bounding boxes, instance segmentations, and other attributes such as velocity and acceleration. Additionally, the dataset includes a detailed map of the environment with over 3,000 km of drivable roads. The dataset is used to train and evaluate perception and prediction models for autonomous vehicles, and provides both raw sensor data and preprocessed data such as object bounding boxes and instance segmentations.

III. Proposed Methodology

A. Network Input

To process the input of six multi-view images, two approaches can be used. The first is to stack the images as separate channels in the neural network input and apply positional encoding to capture the spatial relationships between views. The second approach is to stitch the views into a single image using Homography and apply positional encoding to the resulting image. In this study, we used the first approach and may consider the second approach in future studies.

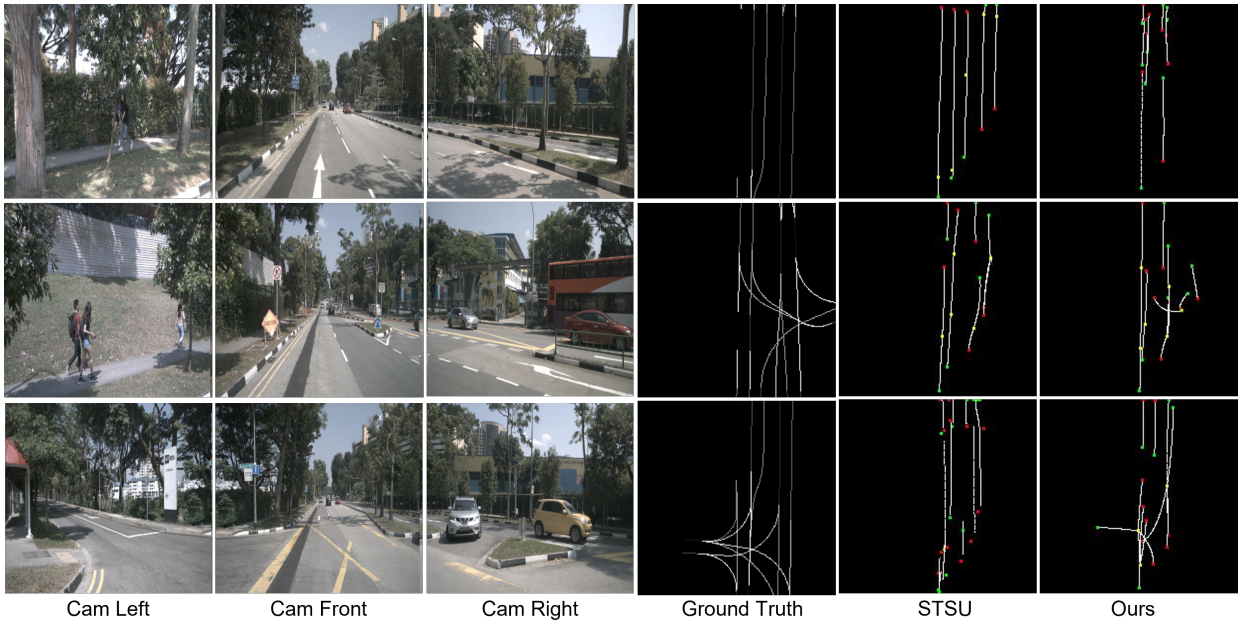

Fig. 3. The centerline estimations for STSU and our method, along with the ground truth, have been compared in various scenarios across 3 samples. (From Left to Right) a) Cam Left image for a sample in a nuScenes scene, b) Cam Front image for a sample in a nuScenes scene, c) Cam Right image for a sample in a nuScenes scene, c) Ground truth image, d) STSU predicted lane centerlines, d) Modified STSU (Our approaqch) predicted lane centerlines

B. Positional Encoding

In [4], positional encoding was introduced for transformers to help the network understand the relative positioning of words. [5] extends positional encoding to the images for the first time. In our case, we need to encode the relationship between pixels from a top-down view. Specifically, by back-projecting pixel rays onto the ground, we obtain BEV coordinates with the camera optical center as the origin as shown in Fig 1(a). Therefore, BEV coordinates can be expressed as a function of image pixel coordinates, as shown in the Eq 1

The proposed approach enhances the BEV grid by adding additional cameras from different perspectives and incorporates camera extrinsics into the positional encoding, unlike earlier methods which only used image pixel coordinates.

In other words, we convert the pixel coordinates to a standard frame of reference (in our case first camera) using the camera extrinsincs provided as shown in Fig 1(b). In equations 1- 4, $\mathcal{F}$ represents the transformation from image coordinates to the BEV coordinates, R, T represent the camera extrinsincs. Finally, equations 3, and 4 represent the sine and cosine based positional encoding which is done in both the approaches.

$$x_{bev}, y_{bev} = \mathcal{F}(u_{img}, v_{img}) \tag{1}$$

$$x_{bev}, y_{bev} = \mathcal{F}(u_{img}, v_{img}, R, T) \tag{2}$$

$$x_{positional\_encoding} = sin(k\pi), cos(k\pi) \tag{3}$$

$$y_{positional\_encoding} = sin(k\pi), cos(k\pi) \tag{4}$$

IV. Implementation Details

A. Experimental Setup

We trained and validated this network on nuScenes dataset as mentioned in section II-A. We used Intel i9 12th generation CPU and 24GB Nvidia 3090ti GPU computer for this work. The total size of the dataset is over 300GB, however, in our experiments, we used only 3 front-facing cameras and generated ground truth centerlines to test the hypothesis of the pipeline. Accordingly, we changed the network hyperparameters, like encoder input vector, embedding vector, and final association vector size to 768 from 256 to accommodate more views. We trained for around 100 epochs with a batch size of 4, as compared to 128 epochs and a batch size of 1 by the original implementation.

B. Ground Truth Generation

The problem with using nuScenes data directly is that the center lines it has are literal lane centers and they don't take traffic elements into account. To circumvent this problem [15] authors refine and create a better lane center line based on Lidar and segmentation labels. Figure 5 summarizes our ground truth generation. The idea is to first get the camera image and the corresponding labels in the image like object masks and lane masks. Parallelly get annotated lane center lines from the nuScenes map extension. Using these data points, [15] authors transform the center lines to the camera frame and mask them based on the occlusion labels. Now, for our purpose, we do the same transformation but run the first part N times for N different views. By doing this we can expand the regenerated ground truth that looks like Figure 4

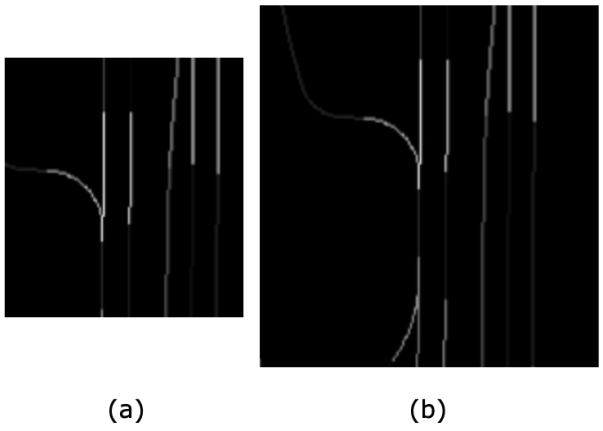

Fig. 4. Comparison between the original and the regenerated ground truth (a) Sample of original ground truth (b) Sample of regenerated ground truth data from the same scene with more views

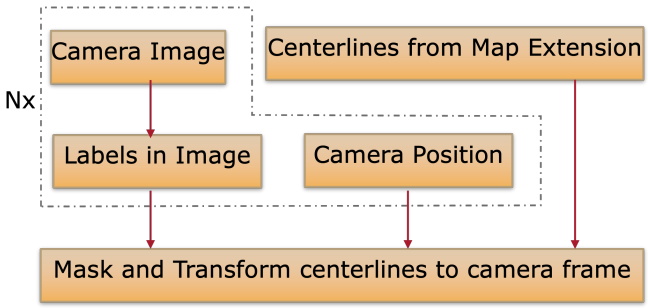

Fig. 5. Ground truth generation: Generating lane centerlines which take occlusions into account. We get the camera image, corresponding labels, and position N_x, once for each camera view

V. Loss and Metrics

Since our approach enhances the existing method developed in [15], we inherit the same loss function and metrics for comparing our results. The loss function is given by $L_m = L_{CE} + \lambda L_1$. Here, L_{CE} is the cross entropy loss for object detection, and L_1 is the 1-norm loss of the Bezier control point locations. Additionally, the metrics described here define how well the estimates fit the real center lines.

Precision-Recall: In this metric, we first match each estimation to the ground truth's (target) Bezier coefficients based on the minimum $L1$ distance. Multiple estimations can be matched with the same target, but each estimation can only be matched with one target. Then, the estimated coefficients are interpolated to obtain dense center lines.

Detection: Detection ratio is used to measure missed center lines which are not measured in the precision-recall metric. The detection ratio is defined as the number of unique GT center lines that at least one estimated line is matched to over the total number of GT center lines.

VI. Results

Figure 3 and table I summarizes our work qualitatively and quantitatively. From figure 3, we feed the "CAM_FRONT" image to the original STSU network, whereas for the modified network we feed all the 3 front-facing camera images. In figure 3, the 2nd and 3rd rows clearly show that our network is able to predict the curved lane center lines near the intersection as compared to STSU. However, detecting straight lane lines is subpar as compared to the STSU. These shortcomings could be attributed to the insufficient training of the network since the new network now has to learn more information. Table I support the qualitative results in the form of less precision-recall. Additionally, our method achieves a higher detection ratio as compared with STSU. This better performance can be ascribed to the predictions from more views.

TABLE I
Comparison of results between STSU and Our Network

Method	M-Prec	M-Rec	Detect
STSU	60.7	54.7	60.6
Ours	52.9	50.8	64.0

VII. Conclusions & Future work

In this work, we propose a method for extracting 3D lane center lines in BEV coordinates from multi-view onboard camera images. We achieve this by enhancing positional embedding for different views by taking the corresponding camera extrinsic parameters. The proposed approach was evaluated against the STSU approach and outperformed in detecting the lanes at intersections giving rise to a high detection ratio.

While our approach shows good results in detecting lane center lines, the approach can be improved further by using more advanced methods like cross-view transformers. Additionally, incorporating additional contextual information, such as road markings or lane boundaries, can further improve the accuracy of our approach to identifying lane center lines.

VIII. Acknowledgements

We would like to thank Prof. Jacob Whitehill for providing this opportunity as part of the course CS 541. We would also like to thank Yigit Baran Can, who is the main author of STSU and has helped us in related development work.

References

[1] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., ... & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3213-3223).

[2] Chen, L., Sima, C., Li, Y., Zheng, Z., Xu, J., Geng, X., Li, H., He, C., Shi, J., Qiao, Y., & Yan, J. (2022). PersFormer: 3D Lane Detection via Perspective Transformer and the OpenLane Benchmark. arXiv preprint arXiv:2203.11089.

[3] Can, Y. B., Liniger, A., Paudel, D. P., & Van Gool, L. (2022). Topology preserving local road network estimation from single onboard camera image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 17263-17272).

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.

[5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[6] Philion, J., & Fidler, S. (2020). Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. arXiv preprint arXiv:2008.05711.

[7] Chen, S., Cheng, T., Wang, X., Meng, W., Zhang, Q., & Liu, W. (2022). Efficient and Robust 2D-to-BEV Representation Learning via Geometry-guided Kernel Transformer. arXiv preprint arXiv:2206.04584.

[8] Zhou, B., & Krähenbühl, P. (2022). Cross-view Transformers for real-time Map-view Semantic Segmentation. arXiv preprint arXiv:2205.02833.

[9] Pham, T., Maghoumi, M., Jiang, W., Jujjavarapu, B. S. S., Liu, M. S. X., Lin, H. C., ... & Park, M. (2023). NVAutoNet: Fast and Accurate 360° 3D Perception For Self Driving. arXiv preprint arXiv:2303.12976.

[10] Van Gansbeke, W., De Brabandere, B., Neven, D., Proesmans, M., & Van Gool, L. (2019). End-to-end lane detection through differentiable least-squares fitting. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 0-0).

[11] Garnett, N., Cohen, R., Pe'er, T., Lahav, R., & Levi, D. (2019). 3d-lanenet: end-to-end 3d multiple lane detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2921-2930).

[12] Efrat, N., Bluvstein, M., Oron, S., Levi, D., Garnett, N., & Shlomo, B. E. (2020). 3d-lanenet+: Anchor free lane detection using a semi-local representation. arXiv preprint arXiv:2011.01535.

[13] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[14] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16 (pp. 213-229). Springer International Publishing.

[15] Can, Y.B., Liniger, A., Paudel, D.P., & Van Gool, L. (2021). Structured Bird's-Eye-View Traffic Scene Understanding from Onboard Images. arXiv preprint arXiv:2110.01997.

[16] Liao, B., Chen, S., Wang, X., Cheng, T., Zhang, Q., Liu, W.,& Huang, C. (2023). MapTR: Structured Modeling and Learning for Online Vectorized HD Map Construction. arXiv preprint arXiv:2208.14437.

[17] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuScenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027.