Fast and Precise Multimodal SpatioTemporal Calibration via Periodic-Activated 2D Gaussian Splatting

Hongbeen Park[1*] Minjeong Park[1*] Sunpil Kim[2] Jinkyu Kim[1] Jung Hyun Lee[2]
[1]Korea University [2]Hyundai Motor Company
{qkrghdqls1, minjeongpark, jinkyukim}@korea.ac.kr {sunpil.kim, jhlee36}@hyundai.com

Abstract

Accurate spatiotemporal calibration between heterogeneous sensors such as cameras and LiDAR is essential for robust performance in tasks like localization, mapping, and object detection. Traditional calibration methods rely on physical targets and manual procedures, limiting scalability and applicability in dynamic or real-time environments. Recent advances in targetless and scene reconstruction-based calibration have improved spatial alignment but often neglect temporal synchronization and are computationally intensive. In this work, we propose PACS (Periodic Activation with 2D Gaussian Splatting for Multimodal SpatioTemporal Calibration), a novel and efficient calibration framework that jointly estimates spatial and temporal alignment across sensors. PACS uses a sine-activated multilayer perceptron (MLP) to effectively capture high-frequency scene details, enhancing convergence and representational power. Leveraging an anchor-based decoding scheme, our method significantly accelerates training while maintaining robust scene reconstruction. Furthermore, we employ 2D Gaussian splatting to render scenes with improved alignment between visual and LiDAR-based geometry. Experimental results demonstrate that PACS achieves accurate, efficient, and robust calibration across multimodal sensor configurations.

1. Introduction

Sensors such as cameras and LiDAR play a critical role in intelligent systems by providing essential spatial information about the surrounding environment. Each sensor type offers a unique modality—for instance, LiDAR delivers precise geometric range data, whereas RGB cameras capture rich color and texture information. Consequently, the fusion of data from multiple sensors enables a more holistic understanding of the scene in the wild, thereby improving the performance of key tasks such as localization [19], mapping [3, 15], and object detection [2, 10, 20]. To fully leverage this multi-modal information from heterogeneous sensors, accurate spatial and temporal calibration is essential.

Sensors frequently operate at different sampling rates, capturing observations at misaligned time steps. This misalignment introduces both temporal and spatial inconsistencies, even when accurate extrinsic parameters are available. Recently, multimodal spatiotemporal calibration has garnered increasing attention, with the goal of jointly estimating both spatial and temporal alignments across heterogeneous sensors. Building on the work of Zhou et al. [21], several studies [5, 7] have proposed NeRF-based pipelines that leverage the differentiable structure of Neural Radiance Fields (NeRF) [13] to jointly optimize scene representations, including geometry, appearance, and sensor poses. While these approaches have demonstrated promising calibration performance, they are computationally demanding and require long training times, limiting their practicality in real-time or large-scale applications.

3DGS-Calib [6] integrates 3D Gaussian Splatting (3DGS) [9] with a multi-resolution hash grid structure [14], enabling progressive learning of scene details from coarse to fine spatial frequencies. Despite the effectiveness, it presents several limitations when applied to large-scale environments such as autonomous driving scenarios [11]. In such cases, representing extensive scenes with sufficient detail requires extremely high-resolution grids, which significantly increases the computational overhead. Moreover, for accurate scene representation, the attributes of neighboring Gaussians often need to vary substantially—especially in regions with complex geometry or fine structures. However, learning such high-frequency variations from grid features using a simple MLP with ReLU activations can be inefficient, resulting in slow convergence.

To this end, we propose PACS (Periodic Activation with 2D Gaussian Splatting for Multimodal SpatioTemporal Calibration), a novel framework for fast and precise multimodal spatiotemporal calibration. The PACS framework is designed to enhance calibration performance across heterogeneous sensors by focusing on three key aspects. First, we

*Equal contribution.

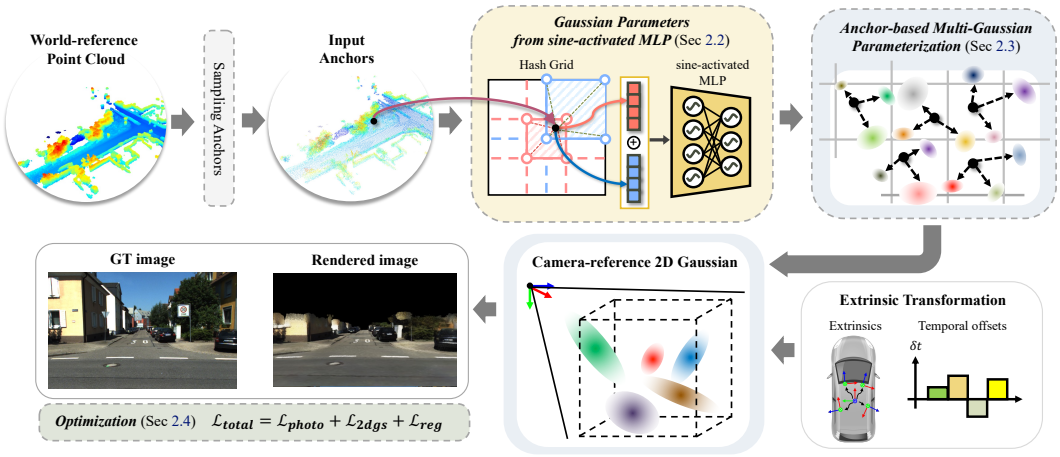


Figure 1. **The pipeline of our framework.** We downsample the accumulated LiDAR point cloud to define anchors. Each anchor generates multiple Gaussians using a hash grid and a sine-activated MLP. These Gaussians are transformed to the camera frame using calibration parameters and rendered via 2D Gaussian Splatting. The rendered image is compared with the ground-truth to compute the photometric loss, and gradients are backpropagated to update both the network and calibration parameters.

introduce a sine-activated multilayer perceptron (MLP) to efficiently capture fine-grained (or high-frequency) visual features using a periodic activation function. Second, the use of the sine-activated MLP enhances the representational power of the network, enabling a reduction in the number of query points through the adoption of an anchor-based decoding scheme [12]. In this scheme, a sparse set of anchor points is utilized to infer the attributes of multiple surrounding Gaussian components. We observe that this approach significantly accelerates the training process while promoting more robust and structured scene reconstruction. Lastly, conventional methods typically employ 3D Gaussians whose positions are fixed to the LiDAR point cloud. This rigid coupling often results in misalignment between the rendered surfaces and the actual LiDAR geometry. To address this limitation, we propose the use of 2D Gaussian splatting for rendering, which enables more precise alignment between the rendered appearance and the surfaces captured by LiDAR. By decoupling the rendering process from fixed 3D positions, our approach achieves improved consistency between visual and geometric modalities.

We validate our approach on the KITTI-360 dataset by injecting noise into the camera extrinsic parameters across multiple sequences. Our method demonstrates strong robustness in recovering accurate spatial and temporal calibration under noisy initialization, while achieving significantly faster convergence compared to prior approaches.

2. Method

2.1. Overall Framework

The spatiotemporal multimodal calibration task estimates extrinsic parameters and temporal offsets of multiple cameras with respect to a reference LiDAR. Given LiDAR poses $\mathbf{T}_t^{\text{LiDAR}} \in \text{SE}(3)$ from SLAM [16] or ICP [1, 4], we interpolate trajectories using SLERP [17] for rotation and linear interpolation for translation, following prior work [5–7]. For each camera C_i, we solve for extrinsics $\Theta_i = \mathbf{R}i, \mathbf{t}i, \delta t_i$, where δt_i is the temporal offset. The global pose at time $t + \delta t_i$ is obtained as

$$\mathbf{T}_{t+\delta t_i}^{C_i} = \mathbf{T}_{t+\delta t_i}^{\text{L}} \cdot \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{1}$$

To represent the scene, we accumulate LiDAR scans into a reference point cloud, downsample to anchors, and use a hash-grid–encoded sine-activated MLP to predict k Gaussian attributes (scale, opacity, rotation, 2D scale). These Gaussians are transformed to the camera frame, rendered via Gaussian Splatting, and optimized against ground-truth images. An overview is shown in Fig. 1.

2.2. Gaussian Parameters from Sine-activated MLP

While conventional Gaussian splatting [8, 9] allows the variation of Gaussian attributes, such as scale and rotation, grid-based optimization with Instant-NGP [14] imposes regularization that encourages similar attribute values among adjacent Gaussians. This regularization is effective for large regions (e.g., walls and roads), however, it limits the ability to capture fine-grained, high-frequency variations inherent in the scene, resulting in slower convergence during optimization. An intuitive solution is to increase the grid resolution; however, it hinders efficient training due to significant computational cost growth.

To balance the efficient cost and performance, we introduce a periodic activation function into the MLP following the grid-based interpolation. Inspired by [18], we use sine activations to enhance the network's ability to represent high-frequency components within the grid, enabling more

expressive Gaussian parameterization without increasing the grid resolution. By leveraging the sine-activated MLP, a small number of anchors can effectively represent a large number of Gaussians while preserving fine details.

2.3. Anchor-based Multi-Gaussian Parameterization

Prior work [6] renders Gaussians from dense point clouds, creating a computational bottleneck. We alleviate this with anchor-based parameterization, generating multiple Gaussians from a single interpolated anchor feature. Since these Gaussians are decoupled from the original point locations, we employ 2D Gaussian Splatting to correct geometric misalignments and ensure accurate alignment.

To generate anchors and initialize Gaussians, we use two point cloud levels: the accumulated dense LiDAR points X and a downsampled cloud X_c. Each point in X_c is an anchor, with k neighbors from X (within its voxel) used to initialize Gaussians. Neighbor coordinates provide initial positional offsets relative to the anchor. If neighbors exceed k, we randomly sample; if fewer, we sample with replacement to ensure a fixed number of Gaussians per anchor.

We denote the learnable positional offset for each Gaussian as $\delta\mathbf{p}$. Given an anchor position $\mathbf{p}_a$, the global position $\mathbf{p}_g$ of a Gaussian is computed as:

$$\mathbf{p}_g = \mathbf{p}_a + \delta\mathbf{p} \tag{2}$$

where $\delta\mathbf{p}$ is initialized with the 3D coordinates of sampled neighboring points. Other Gaussian attributes—scale, opacity, color, and rotation—are predicted through separate output heads of the neural network. Each head predicts k sets of attribute values for each anchor, resulting in an output tensor of shape $N \times d \times k$, where N is the number of anchors, d is the dimensionality of each attribute (e.g., 3 for color), and k is the number of Gaussians per anchor. This design enables efficient parallel prediction of multiple Gaussians from each anchor feature while maintaining modularity across different attributes. Although using only N sparse anchors as input, our method produces NK dense Gaussians, enriching scene representation without extra feature extraction cost.

2D Gaussian Splatting for Scene Representation. Since the positions of the Gaussians are learnable and may deviate from the original LiDAR points during optimization, maintaining geometric consistency with the underlying LiDAR structure becomes challenging. To address this issue, we introduce a depth consistency loss to enforce alignment between the generated Gaussians and the LiDAR point cloud. The depth loss is defined as:

$$\mathcal{L}_{depth} = ||D(\Phi, \Theta_i, \delta\mathbf{p}) - \bar{D}(\Theta_i)||_1 \tag{3}$$

where $\bar{D}(\Theta_i)$ denotes the depth map obtained by projecting the original point cloud X using the given camera extrinsic transformation Θ_i, and $D(\Phi, \Theta_i, \delta\mathbf{p})$ represents the depth map rendered from the Gaussians with the learnable offsets. By minimizing the discrepancy between these two depth maps, we prevent misalignments between the Gaussian representations and the original LiDAR geometry, ensuring accurate spatial consistency during training. To ensure that the optimization focuses solely on adjusting the offsets, we detach the neural representation Φ and the camera extrinsic parameters Θ_i from the computation graph when computing the depth loss. This prevents gradients from being propagated to Φ and Θ_i , ensuring that only $\delta\mathbf{p}$ is updated by the depth supervision.

2.4. Optimization

Rendered Image Optimization. The Gaussians are splatted to render an image, and the photometric loss $\mathcal{L}_{photo}$ minimizes the discrepancy between the rendered and GT image. Specifically, following the formulation from [9], $\mathcal{L}_{photo}$ consists of an $L1$ loss and a D-SSIM term:

$$\mathcal{L}_{photo} = (1 - \lambda_{ssim})\mathcal{L}_1 + \lambda_{ssim}\mathcal{L}_{\text{D-SSIM}} \tag{4}$$

2D Gaussians Optimization. Following [8], we introduce 2DGS optimization function to improve geometric consistency and stability for 2D Gaussians:

$$\mathcal{L}_{2dgs} = \lambda_d\mathcal{L}_d + \lambda_n\mathcal{L}_n \tag{5}$$

Regularization Terms. To stabilize the optimization process, we propose two additional regularization terms: a scale normalization loss $\mathcal{L}sn$ to prevent overly sharp Gaussians that harm stability and rendering quality, and an offset loss $\mathcal{L}os$ to limit drift from anchor positions, promoting geometric consistency while allowing flexible offsets. These losses are defined as:

$$\mathcal{L}_{sn} = \frac{1}{N \cdot k}\sum_{i}^{N \cdot k}||s_i - \tilde{s}_i||, \quad \mathcal{L}_{os} = \frac{1}{N \cdot k}\sum_{i=1}^{N}\sum_{j=1}^{k}\left|\delta\mathbf{p}_{i,j}\right|_1 \tag{6}$$

The proposed regularization loss $\mathcal{L}_{reg}$ is defined as:

$$\mathcal{L}_{reg} = \lambda_{depth}\mathcal{L}_{depth} + \lambda_{sn}\mathcal{L}_{sn} + \lambda_{os}\mathcal{L}_{os} \tag{7}$$

Finally, during the training phase, we employ the loss functions defined above as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{photo} + \mathcal{L}_{2dgs} + \mathcal{L}_{reg} \tag{8}$$

3. Experimental Results

Spatiotemporal calibration. We evaluate both spatial and temporal calibration results in Tab. 1. While the calibration errors of prior methods vary across different scenes, our method consistently achieves less than 0.21° in rotation, 9.8 cm in translation, and 7.1 ms in time offset. In

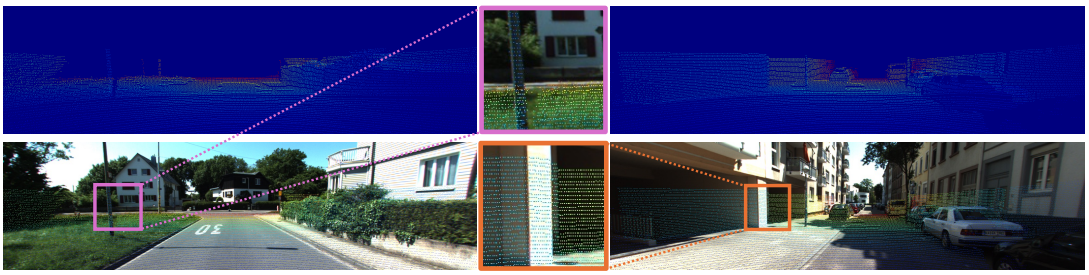
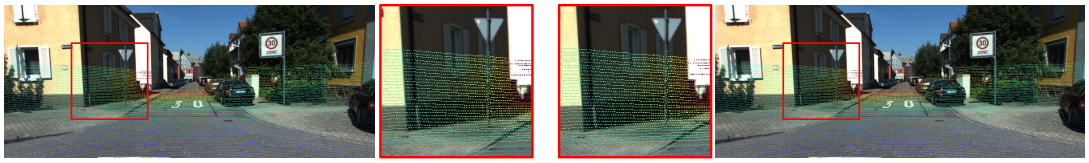
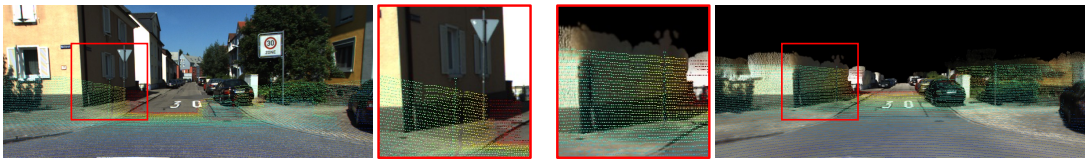


Figure 2. **Calibration results of PACS.** Boxes with the same color indicate zoomed-in views of the corresponding regions.



(a) GT (RGB Image) (c) Ours (RGB Image)



(b) Initial errors (RGB Image) (d) Ours (Rendered Image)

Figure 3. **Comparison of calibration results. (a) GT on RGB image, (b) Initial error on RGB image, (c) Ours on RGB image, and (d) Ours on rendered image on KITTI-360 [11] dataset.** The red boxes show enlarged views of the same regions across the images.

Method	Voxel Size	Training time (sec) ↓	Rotation (°) ↓	Translation (cm) ↓	Time offset (sec) ↓	#. of Gaussians
3DGS-Calib [6]	0.10	285	0.28	14.1	8.3	374,797
3DGS-Calib [6]	0.05	620	0.28	10.1	7.2	1,210,066
3DGS-Calib [6]	0.02	1390	0.38	**9.0**	7.9	3,196,593
3DGS-Calib [6]	Iteration-dependent	490	0.31	10.3	**6.7**	Voxel-dependent
Ours (PACS)	0.10	**264**	**0.21**	9.8	7.1	1,873,985

Table 1. **Comparison of Spatiotemporal Calibration Accuracy with 3DGS-Calib [6].** 'Iteration-dependent' represents the voxel size is dependent on the iteration number, and 'Voxel-dependent' denotes that the number of Gaussians in 3DGS-Calib depends on the voxel sizes.

addition, our method achieves comparable calibration accuracy to 3DGS-Calib while requiring significantly less training time. Furthermore, while our method also uses voxel-based downsampling to define anchors, it does not require dynamic tuning of the voxel resolution as in 3DGS-Calib. This allows efficient training with a large number of Gaussians without costly resolution adjustment.

Calibration results of PACS. In Figure 2, we visualize the alignment of reprojected 3D LiDAR points onto the front-facing image as a result of our LiDAR-to-image calibration. As shown in the figure, our method successfully aligns the LiDAR point cloud with the camera image, indicating accurate spatial calibration between the two sensors. Additionally, we reproject the 3D points onto a rendered image that is generated using the same estimated pose as the LiDAR mounted on the vehicle.

Comparison of calibration results. We visualize the calibration process from the initial state (e.g., (b)) to the final output (e.g., (c), (d)) and compare it with the ground truth (e.g., (a)) to demonstrate the effectiveness of our training. As shown in Figure 3, our method accurately performs calibration (see (b), (c)). Following MOISST [5], we project 3D points onto a synthetic image rendered from the LiDAR viewpoint to ensure alignment and avoid parallax distortions (see (d)). These results highlight the precise spatial alignment achieved even in complex scenes with fine geometric details.

4. Conclusion

We propose an efficient multimodal spatiotemporal calibration framework using 2D Gaussian Splatting. A sine-activated MLP enhances high-frequency detail and accelerates convergence, while an anchor-based decoding scheme infers Gaussian attributes from sparse anchors, reducing computation without sacrificing fidelity. To address geometric misalignment of 3D Gaussians, we adopt 2D splatting for precise alignment with LiDAR. Experiments demonstrate both efficiency and accuracy in real-world dynamic settings.

Acknowledgements. This work was supported by Autonomous Driving Center, Hyundai Motor Company R&D Division. This work was also partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the artificial intelligence star fellowship support program to nurture the best talents (IITP-2025-RS-2025-02304828) grant funded by the Korea government(MSIT).

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987. 2

[2] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan 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*, pages 1090–1099, 2022. 1

[3] Jaewoong Choi, Junyoung Lee, Dongwook Kim, Giacomo Soprani, Pietro Cerri, Alberto Broggi, and Kyongsu Yi. Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment. *IEEE Transactions on Intelligent Transportation Systems*, 13(2): 974–982, 2012. 1

[4] Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. Rgbd gs-icp slam. In *European Conference on Computer Vision*, pages 180–197. Springer, 2024. 2

[5] Quentin Herau, Nathan Piasco, Moussab Bennehar, Luis Roldão, Dzmitry Tsishkou, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. Moisst: Multimodal optimization of implicit scene for spatiotemporal calibration. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1810–1817. IEEE, 2023. 1, 2, 4

[6] Quentin Herau, Moussab Bennehar, Arthur Moreau, Nathan Piasco, Luis Roldão, Dzmitry Tsishkou, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. 3dgs-calib: 3d gaussian splatting for multimodal spatiotemporal calibration. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8315–8321. IEEE, 2024. 1, 3, 4

[7] Quentin Herau, Nathan Piasco, Moussab Bennehar, Luis Roldao, Dzmitry Tsishkou, Cyrille Migniot, Pascal Vasseur, and Cédric Demonceaux. Soac: Spatio-temporal overlap-aware multi-sensor calibration using neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15131–15140, 2024. 1, 2

[8] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 2, 3

[9] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Gr.*, 42(4):139–1, 2023. 1, 2, 3

[10] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc 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*, pages 17182–17191, 2022. 1

[11] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2023. 1, 4

[12] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2

[13] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 1

[14] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 1, 2

[15] Hongbeen Park, Minjeong Park, Giljoo Nam, and Jinkyu Kim. Lrslam: Low-rank representation of signed distance fields in dense visual slam system. In *European Conference on Computer Vision*, pages 225–240. Springer, 2024. 1

[16] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765, 2018. 2

[17] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, page 245–254, New York, NY, USA, 1985. Association for Computing Machinery. 2

[18] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 2

[19] Rafael Peixoto Derenzi Vivacqua, Massimo Bertozzi, Pietro Cerri, Felipe Nascimento Martins, and Raquel Frizera Vassallo. Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 19 (2):582–597, 2017. 1

[20] Jin Hyeok Yoo, Yecheol Kim, Jisong Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XXVII 16*, pages 720–736. Springer, 2020. 1

[21] Shuyi Zhou, Shuxiang Xie, Ryoichi Ishikawa, Ken Sakurada, Masaki Onishi, and Takeshi Oishi. Inf: Implicit neural fu-

sion for lidar and camera. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10918–10925. IEEE, 2023. 1