

DENSER: 3D Gaussians Splatting for Scene Reconstruction of Dynamic Urban Environments

Mahmud A. Mohamad, Gamal Elghazaly, Arthur Hubert, and Raphael Frank

Abstract—This paper presents DENSER, an efficient and effective approach leveraging 3D Gaussian splatting (3DGS) for the reconstruction of dynamic urban environments. While several methods for photorealistic scene representations, both implicitly using neural radiance fields (NeRF) and explicitly using 3DGS have shown promising results in scene reconstruction of relatively complex dynamic scenes, modeling the dynamic appearance of foreground objects tend to be challenging, limiting the applicability of these methods to capture subtleties and details of the scenes, especially far dynamic objects. To this end, we propose DENSER, a framework that significantly enhances the representation of dynamic objects and accurately models the appearance of dynamic objects in the driving scene. Instead of directly using Spherical Harmonics (SH) to model the appearance of dynamic objects, we introduce and integrate a new method aiming at dynamically estimating SH bases using wavelets, resulting in better representation of dynamic objects appearance in both space and time. Besides object appearance, DENSER enhances object shape representation through densification of its point cloud across multiple scene frames, resulting in faster convergence of model training. Extensive evaluations on KITTI dataset show that the proposed approach significantly outperforms state-of-the-art methods by a wide margin. Source codes and models will be uploaded to this repository <https://github.com/sntubix/denser>

I. INTRODUCTION

Modeling dynamic 3D urban environments from images has a wide range of important applications, including building city-scale digital twins and simulation environments that can significantly reduce training and testing costs of autonomous driving systems. These applications demand efficient and high-fidelity 3D representation of the road environment from captured data and the ability to render high-quality novel views in real time. Simulation is crucial for developing and refining autonomous driving functions by providing a controlled, safe, and cost-effective testing environment. While traditional simulation tools like CARLA [1], LGSVL [2], and DeepDrive [3] have accelerated autonomous driving development they all share of common limitation, a large *sim-to-reality* gap [4]. This gap is induced by the limitations in asset modelling and rendering that hinder model-based simulation tools their ability to fully replicate the complexities of the real world.

Mahmud A. Mohamad, Gamal Elghazaly, Arthur Hubert and Raphael Frank are with SNT - Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, 29 Avenue John F. Kennedy, L-1855 Luxembourg, Luxembourg. Email: {mahmud.ali, gamal.elghazaly, arthur.hubert, raphael.frank}@uni.lu



Fig. 1. Scene decomposition using DENSER into static background and dynamic objects and reconstruction (a) Ground truth (b) scene decomposition: static background (c) scene decomposition: dynamic objects (d) scene reconstruction

To close this gap, new data-driven and photorealistic techniques based NeRFs [5] and 3DGS [6] have shown significant capabilities for 3D scene reconstructions to visually and geometrically realistic fidelity. While NeRFs and 3DGS excel in static and small-scale scene reconstruction, reconstructing highly dynamic and complex large urban scenes remains a significant challenge.

Both NeRFs [5] and 3DGS [6] represent two distinct approaches that have shown ground-breaking scene representation results enabling photorealistic rendering and synthesising novel views of the 3D scene. While NeRFs *implicitly* represent the radiance field and density of the 3D scene, 3DGS *explicitly* represent the scene using a large set anisotropic 3D Gaussians with associated color and opacity features. This explicit of 3DGS representation results in a faster training and rendering compared to NeRFs, thanks to parallel rasterization computed in GPUs. Despite the significant potential of NeRFs and 3DGSs in static scene representation, their performance deteriorates considerably in dynamic scenes involving moving transient objects or when faced with changing conditions such as weather, exposure, and varying lighting [7], [8]. Numerous works have already attempted to address this challenge. Early approaches disregarded dynamic objects and focused solely on reconstructing static components of the scene [9], [7], [8], [10], rendered viewers from these approaches typically suffer from artefacts induced by transient objects. Two different approaches for dynamic scene representation have shown initial but promising results. The first represents the scene the scenes as a combination of a static and time-varying radiance fields [11], [12], [13]. In the second approach, graph is used to represent the scene and its nodes represent static background and foreground dynamics objects, while edges maintain relationships between scene static and dynamic entities needed for scene composition

over time [14], [15], [16], [17]. However, most of these scene graph-based approaches do not or insufficiently consider the appearance of dynamic objects over time. This paper proposes DENSER, a scene graph-based framework that significantly enhances the representation of dynamic objects and accurately models the appearance of dynamic objects in the driving scene (Fig. 1). Instead of directly using Spherical Harmonics (SH) to model the appearance of dynamic objects, we introduce and integrate a new method aiming at dynamically estimating SH bases using wavelets, resulting in a better representation of dynamic objects appearance in both space and time. Our proposed methods achieve superior scene decomposition on the KITTI dataset.

The rest of this paper is organized as follows. Section II provides a review of related work in 3D scene reconstruction. Section III presents the proposed methodology. Section IV presents experimental results, demonstrating the effectiveness of our approach on the KITTI dataset. Finally, Section V concludes this paper.

II. RELATED WORK

Dynamic scene representation has seen remarkable progress, especially in the domain of 4D neural scene representations focusing on scenes of single dynamic object, where time is considered as an additional dimension besides spatial ones [18], [19], [20], [21], [22], [23], [24]. Alternative to time modulation, dynamic scenes can be modelled by coupling a deformation network to map time-varying observations to canonical deformations [25], [26], [27]. These approaches are generally limited to small-scale scenes and slight movements and are considered inadequate for complex urban environments. Furthermore, these approaches are not designed to decouple dynamic scenes into their static and dynamic primitives, e.g. instance-aware decomposition, therefore their applicability in autonomous driving simulations is limited. Alternatively, explicit decomposition of the dynamic scene facilitates accessibility and editing to manipulate these objects for simulation purposes. Scene graph has been used to model the relations between the entities composing the scene as in Neural Scene Graphs (NSG) [17], MARS [14], UniSim [28], StreetGaussians [15], and [16]. However, scene graph-based methods handle objects with limited time-varying appearances. This paper uses wavelets to enhance scene graph-based methods and how to model accurately models the appearance of dynamic objects in the driving scene.

III. FRAMEWORK AND METHODOLOGY

A. Preliminaries

As introduced in [6], 3DGS represents a scene explicitly using a finite set of 3D n anisotropic Gaussians $\mathcal{G} = \{\mathcal{G}_i\}$, each is defined by a 5-tuple $\mathcal{G}_i = \langle \mu, S, R, \alpha, c \rangle, \forall i = 1, 2, \dots, n$, where $\mu \in \mathbb{R}^3$ represents its centroid, $S \in \mathbb{R}_{+}^3$ is a scale vector, $R \in SO(3)$ its rotation matrix, $\alpha \in (0, 1)$ is opacity, and $c \in \mathbb{C}^3$ is a view-dependent color, often

represented using a set of coefficients in a basis of SH. The 3D volume G_i occupied by the Gaussian \mathcal{G}_i could be expressed as

$$G_i(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (1)$$

The covariance matrix Σ of \mathcal{G}_i could be decomposed using the rotation matrix R and the scale vector S as

$$\Sigma = RSS^T R^T \quad (2)$$

For rendering, these 3D Gaussians are projected to 2D, and their covariance matrices are transformed accordingly. This involves computing a new covariance matrix Σ' in camera coordinates using the Jacobian of the affine approximation of the projective transformation J and a viewing transformation W [29]

$$\Sigma' = JW\Sigma W^T J^T \quad (3)$$

To compute the color c of a pixel is calculated using an N -ordered 2D splats using α -blending

$$c = \sum_{i=1}^N c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (4)$$

While 3DGS performs well in static and object-centric small scenes, it faces challenges when dealing with scenes featuring transient objects and varying appearances [?]. This paper proposes a framework to model the appearance of dynamic objects by dynamically estimating the SH coefficients using wavelets, resulting in better representation of dynamic objects appearance in both space and time.

B. Scene Graph Representation

As shown in Fig. 2, the proposed framework is built on a scene graph representation accommodating both static background and dynamic objects. In DENSER, the scene is decomposed into *background node* representing the static entities in the environment such roads and buildings and *object nodes*, each represent a dynamic object in the scene, e.g. vehicles. Each of these nodes are represented using a set of 3D Gaussians as described in Section III-A that are optimized separately for each node. While the background node is directly optimized in the world reference frame \mathcal{W} , the object nodes are optimized in their object reference frame \mathcal{O}_i that can be transformed into the world reference frame. All Gaussians corresponding to both background node and dynamic objects nodes are concatenated for rendering in a similar manner as proposed in [15], [6], [16].

Let us denote $\mathcal{G}_b^{\mathcal{W}}$ as the set of 3D Gaussians representing the background node and $\mathcal{G}_i^{\mathcal{O}}$ as the set of 3D Gaussians representing dynamic object i in its object reference frame \mathcal{O}_i . Given the trajectory $\tau_i : t \rightarrow T$ of object i , one can extract a pose transformation matrix $T_i^{\mathcal{W}}(t) \in SE(3)$ representing the position and orientation of object i at time t . Assuming the geometry of objects does not change from one pose to another, one can

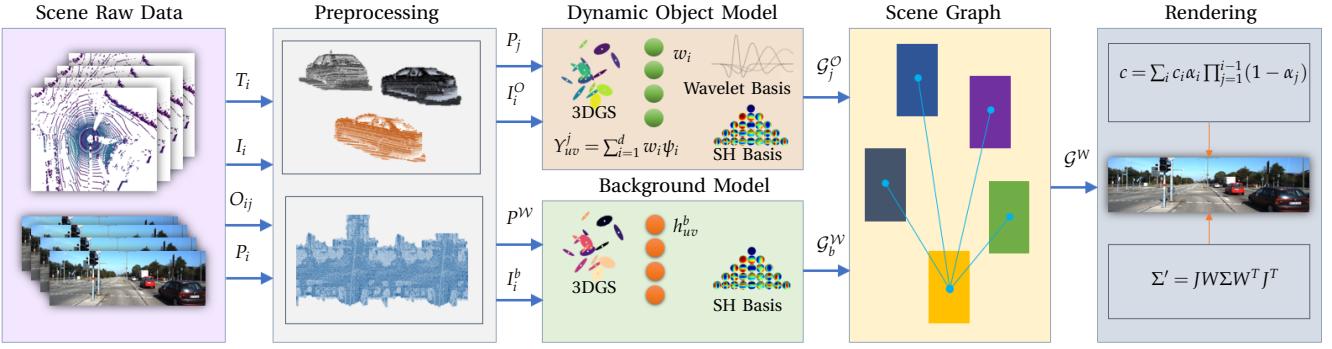


Fig. 2. DENSER Scene Composition Framework. The pipeline starts by processing raw sensor data to get a set of densified point cloud for each foreground object in its reference frame and for the static background. Object point clouds are used to initialize 3D Gaussians of dynamic objects for which wavelets are used to estimate their color appearance. Background point cloud initializes the 3D Gaussians of the static with appearance modelled using a traditional SH basis. All 3D Gaussians form a scene graph which can jointly rendered for a novel view.

simply transform $\mathcal{G}_i^{\mathcal{O}}$ to the word frame by applying homogeneous transformation using $T_i^{\mathcal{W}}(t)$ as follows

$$\mathcal{G}_i^{\mathcal{W}}(t) = T_i^{\mathcal{W}}(t) \otimes \mathcal{G}_i^{\mathcal{O}} \quad (5)$$

The set of all Gaussians to be used for rendering can be obtained by concatenating all sets Gaussians of the static background node and transformed dynamic objects node

$$\mathcal{G}^{\mathcal{W}} = \bigoplus_{j=0}^m \mathcal{G}_j^{\mathcal{W}}, \quad \forall j = 0, 1, 2, \dots, m, \quad (6)$$

with $j = 0$ represents the background, i.e. $\mathcal{G}_b^{\mathcal{W}} = \mathcal{G}_0^{\mathcal{W}}$ and the remaining sets of Gaussians are those of dynamic object nodes.

C. Scene Decomposition

This paper improves existing 3DGS composite scene reconstruction by enhancing the modeling of appearance of transient objects, resulting in a more realistic and consistent scene representation. The input to DENSER is a sequence n frames. The frame \mathcal{F}_i is defined in term of a set of m tracked objects, a sensor pose T_i , a LIDAR point cloud P_i and a set camera images I_i and optionally a depth map D_i , $\forall i \in \{1, 2, \dots, n\}$. Each object j in the frame i , O_{ij} is often defined by a bounding box, a tracking identifier, and an object class, $\forall j \in \{1, 2, \dots, m\}$. Based on these inputs, DENSER starts by accumulating point clouds from over all frames in the world frame \mathcal{W} while using object bounding boxes to filter the points corresponding to foreground objects. The resulting point cloud $P_b^{\mathcal{W}}$ is to initialize the 3D Gaussians of the background $\mathcal{G}_b^{\mathcal{W}}$ for the position μ_b , opacity α_b and covariance Σ_b and the corresponding rotation R_b and scale S_b as described in (2) in a similar to [6]. Besides, each Gaussian of the background is assigned a set of SH coefficients $H^b = \{h_{uv}^b \mid 0 \leq u \leq U, -u \leq v \leq V\}$, where U and V are defined by the order of SH basis defining the view-dependent color $Y_{uv}^b(\theta, \phi)$, with θ and ϕ define the viewing direction. While for static scenes, the original 3DGS has shown to be capable of representing scene efficiently, it struggles to represent scenes including dynamic entities

and varying appearances [7]. Representing the appearance of transient objects solely using SH coefficients tends to be insufficient [15]. This arises mainly from the sensitivity of SH to the changes in the position of the objects in the scene and the associated changes in shadows and lighting induced by these motions. To maintain a consistent visual appearance, DENSER handles this challenge by using (i) densification of object point clouds across all different frames, which ensures not only a strong prior for initialization of the 3D Gaussians, but also mitigates the pose calibration errors and noisy measurements inherent in the datasets. Using the sensor pose transformation matrix T_j and LIDAR point cloud P_i , one can apply an ROI filter defined using the bounding box of the object O_j to get the point cloud P_{ij} of object j at frame i . Concatenating across all frames results in the densified point cloud P_j^d used for initialization. (ii) We use a time-dependent approximation of SH bases to capture the varying appearance of dynamic objects using an orthonormal basis of wavelets with scale and translation parameters are optimizable parameters. In DENSER, the Ricker wavelet is used

$$\psi(t) = \frac{2}{\sqrt{3a}\pi^{1/4}} \left(1 - \left(\frac{\tau}{a} \right)^2 \right) \exp \left(-\frac{\tau^2}{2a^2} \right), \quad (7)$$

where a is its scale parameter and $\tau = t - b$, with b is its translation parameter. The SH basis function $Y_{uv}^i(\theta, \phi)$ for object j is approximated using the linear combination of child wavelets

$$Y_{uv}^j(t) = \sum_{i=1}^d w_i \psi(t, a_i, b_i) \quad (8)$$

where d is the dimension of the wavelet basis and w_i is also an optimizable parameters. Unlike the truncated Fourier transform used in [15], wavelets are known to capture higher frequency contents even with a finite dimension of wavelet basis, resulting in significant performance to capture dynamic object details as well as varying appearances. Both (i) and (ii) constitute the genuine contribution of the present paper.

D. Optimization

To optimize our scene, we employ a composite loss function \mathcal{L} defined as

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{accum}}, \quad (9)$$

where $\mathcal{L}_{\text{color}}$ represents the reconstruction loss to ensure that the predicted image I_{pred} closely matches the GT image I_{gt} . This is achieved through a combination of \mathcal{L}_1 loss and Structural Similarity Index (SSIM) loss. The \mathcal{L}_1 loss is given by $\mathcal{L}_1 = \|I_{\text{gt}} - I_{\text{pred}}\|_1$ and the SSIM loss $\mathcal{L}_{\text{SSIM}}$ is given by $\mathcal{L}_{\text{SSIM}} = 1 - \text{SSIM}(I_{\text{gt}}, I_{\text{pred}})$ with $\mathcal{L}_{\text{SSIM}}$ quantifies the similarity between two images, taking into account changes in luminance, contrast, and structure. SSIM evaluates image quality and is more sensitive to structural information. The total color loss $\mathcal{L}_{\text{color}}$ is defined in terms of \mathcal{L}_1 and $\mathcal{L}_{\text{SSIM}}$ as $\mathcal{L}_{\text{color}} = (1 - \lambda_c)\mathcal{L}_1 + \lambda_c\mathcal{L}_{\text{SSIM}}$ where λ_c is a parameter to encourage structural alignment between I_{gt} and I_{pred} [6]. $\mathcal{L}_{\text{depth}}$ is the mono-depth loss, which ensures that the predicted depth maps are consistent with the observed depth information. This term helps maintain the geometric consistency of the scene. The depth loss $\mathcal{L}_{\text{depth}}$ is computed as the \mathcal{L}_1 loss between the predicted depth D_{pred} and the ground truth depth D_{gt} maps as $\mathcal{L}_{\text{depth}} = \lambda_d\|D_{\text{gt}} - D_{\text{pred}}\|_1$ and $\mathcal{L}_{\text{accum}}$ is the accumulation loss, which penalizes the deviation of accumulated object occupancy probabilities from the desired distributions. Specifically, it includes an entropy-based loss to ensure balanced occupancy probabilities for each object as $\mathcal{L}_{\text{accum}} = -(\beta \log(\beta) + (1 - \beta) \log(1 - \beta))$ where β represents the object occupancy probability. This composite loss function facilitates the simultaneous optimization of appearance, geometry, and occupancy probabilities, ensuring a coherent and realistic reconstruction of the scene.

IV. EXPERIMENTS AND RESULTS

A. Dataset and Baselines

We conduct comprehensive evaluation of DENSER for reconstructing dynamic scenes on the KITTI dataset [30] as one of the standard benchmark for scene reconstructions in urban environments. Data frames in KITTI are recorded at 10Hz. We follow the same settings and evaluation methods used in NSG [17], MARS [14] and StreetGaussians [15] which constitute the recent methods we use as our baseline for quantitative and qualitative comparisons.

B. Implementation Details

The training setup for our scene reconstruction utilizes the Adam optimizer across all parameters, with 30K iterations. The learning rate for the wavelets scale and translation parameters is set to $r = 0.001$ with $\epsilon = 1 \times 10^{-15}$. All experiments are conducted on an NVIDIA Tesla V100-SXM2-16GB GPU. In our comparative analysis, we observed that NSG [17] and MARS [14] trained their models for 200K and 350K iterations, respectively, while the Street Gaussian [15] reported training for 30K iterations. To determine the optimal training regimen,

we tested all these configurations and found that the improvement in reconstruction quality was negligible beyond 30K iterations, with a gain of only about 0.2 in PSNR when extending from 30K to 350K iterations. Given the minimal improvement and the significant increase in training time, extending to 350K iterations was not justifiable. Specifically, training for 30K iterations takes approximately 30 minutes, whereas 350K iterations would require around 5.0 hours.

C. Results and Evaluation

We conduct qualitative and quantitative comparisons against other state-of-the-art methods. These methods include, NSG [17], which represents the background as multi-plane images and utilizes per-object learned latent codes with a shared decoder to model moving objects. MARS [14], which builds the scene graph based on Nerfstudio [31], 3D Gaussians [6], which models the scene with a set of anisotropic Gaussians, and StreetGaussian [15], which represents the scene as composite 3D Gaussians for foreground and background representation. We directly use the metrics reported in their respective papers to compare against our method. Table I, presents the quantitative comparison results of our method with baseline methods. As we strictly followed the same procedure and settings used in MARS and StreetGaussians (SG) to legitimately borrow their results for comparison. The rendering image resolution is 1242×375. Our approach significantly outperforms previous methods. The training and testing image sets in the image reconstruction setting are identical, whereas in novel view synthesis, we render frames that are not included in the training data. Specifically, we hold out one in every four frames for the 75% split, one in every two frames for the 50% split, and only every fourth frame is used for training in the 25% split, resulting in 25%, 50%, and 75% of the data being used for training, respectively. We adopt PSNR, SSIM, and LPIPS as metrics to evaluate rendering quality. Our model achieves the best performance across all metrics. Our experimental results indicate that DENSER performs exceptionally well in reconstructing dynamic scenes compared baseline methods. The results show significant improvements in Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) metrics, as detailed in Table I. The improvements in PSNR and SSIM highlight our wavelet-based approach's effectiveness in maintaining high fidelity and structural integrity in complex environments. Furthermore, DENSER has shown to be capable of reconstructing little details, e.g. shadow at the back of truck in Scene 0006 as shown in Fig. 3, while other baseline methods are not.

D. Ablation on the Dimension of Wavelet Basis

We conducted an ablation study to analyse the impact of the size of the wavelet basis, e.g. the number of wavelets used to approximate the SH functions. We run our experiments while incrementing the dimension of wavelets

TABLE I

Quantitative results on KITTI [30] comparing our approach with baseline methods, MARS [14], SG [15], NSG [17], and 3DGS [6]

	KITTI - 75%			KITTI - 50%			KITTI - 25%		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS [6]	19.19	0.737	0.172	19.23	0.739	0.174	19.06	0.730	0.180
NSG [17]	21.53	0.673	0.254	21.26	0.659	0.266	20.00	0.632	0.281
MARS [14]	24.23	0.845	0.160	24.00	0.801	0.164	23.23	0.756	0.177
SG [15]	25.79	0.844	0.081	25.52	0.841	0.084	24.53	0.824	0.090
Ours	31.73	0.949	0.025	31.19	0.945	0.027	30.408	0.935	0.031

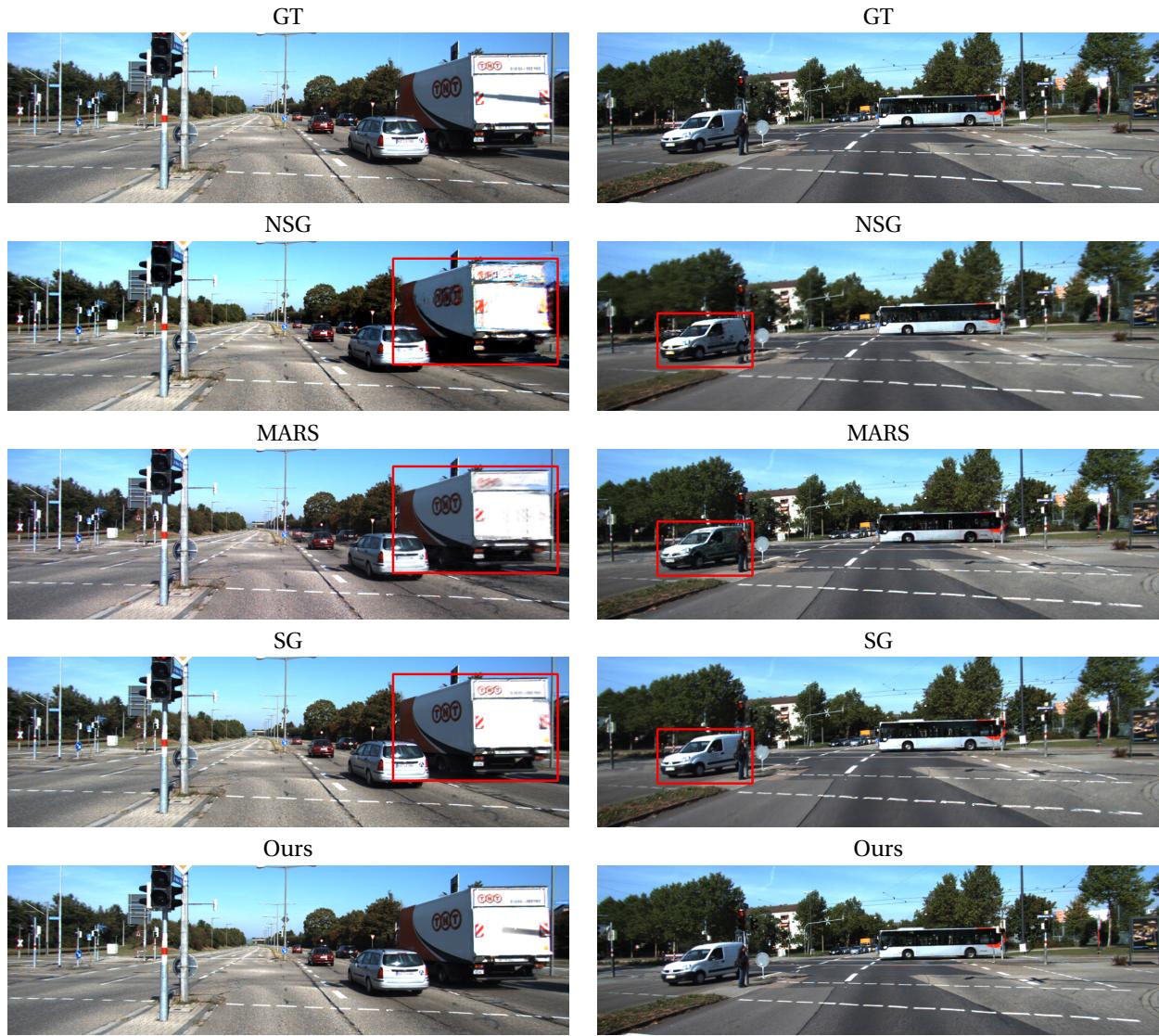


Fig. 3. Qualitative image reconstruction comparison on KITTI dataset [30].

and analysing the impact on performance metrics (PSNR↑, SSIM↑ LPIPS↓), we used for evaluation in order obtain the optimal dimension giving the best performance. As shown in Fig. 4, the performance increases gradually up to 7 wavelets and starts to degrade gradually after it.

E. Scene Editing Applications

DENSER enables photorealistic scene editing, such as swapping, translating, and rotating vehicles, to create diverse and realistic scenarios. This versatility allows autonomous systems to improve their performance and their ability to handle complex real-world conditions, from routine traffic to critical situations.

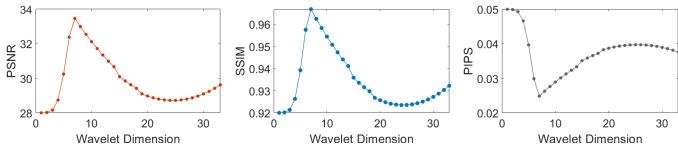


Fig. 4. Ablation: Impact of the dimension of wavelet basis on the performance of scene reconstruction

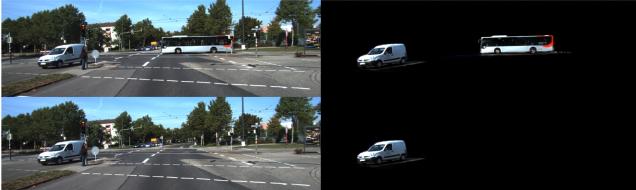


Fig. 5. **Object Removal:** The top row shows the GT while the bottom row displays the modified scenes where the bus have been removed.

1) *Object Removal*: To remove an object, we simply construct a deletion mask that effectively filters out the Gaussian parameters associated with the objects to be removed. The deletion mask is then applied to the Gaussian parameters of the trained model, removing the attributes associated with the unwanted objects as shown in Fig. 5.

2) *Object swapping*: Swapping vehicles within our representational framework is a straightforward process that involves a simple exchange of unique track ids associated with the two target vehicles. This manipulation results in a dynamic alteration of the scene, wherein a vehicle assumes the spatial attributes, specifically location and orientation, of the vehicle with which it has been swapped as depicted in Fig. 6.



Fig. 6. **Object Swapping:** The top shows the GT. In the bottom, the two vehicles within the red box of the top image have been replaced with different ones in the bottom, and some vehicles have been removed for better visualization

3) *Object Rotation and Translation*: Translation and orientation modifications are implemented to adjust an object's position and heading dynamically within a 3D environment. Given an object position rotation matrix at a specific timestep i , we can modify the translation and rotation to achieve desired motion maneuver. For the sake of illustration in this paper, one can shift the translation component in the plan of motion to achieve translation,



Fig. 7. **Rotation and Translation:** The top row displays GT, illustrating the original positions and orientations of the vehicles. In the middle and bottom left images, the vehicles have been rotated. In the middle and bottom right images, the vehicle has been both rotated and translated to another lane.



Fig. 8. **Trajectory Alteration:** The left column displays the GT trajectory and right column shows the vehicle follows a new modified path.

while for rotation, we can transform change rotation angle about the normal to the plan of motion and calculate back the corresponding new rotation matrix to be used to replace the object as depicted in Fig. 7.

4) *Trajectory Alteration*: A trajectory is defined as a sequence of poses. Editing the scene to allow an object follow a trajectory, one can generalize the change in rotation and translation not only between two configurations as previously illustrated but to apply this change over time to obtain a smooth change in translation and rotation as a function of time as illustrated in Fig. 8.

V. CONCLUSION

In this paper, we presented DENSER, a novel and efficient framework leveraging 3DGS for reconstruction of dynamic urban environments. By addressing the limitations of existing methods in modeling the appearance of dynamic objects, particularly in complex driving scenes, DENSER demonstrates significant improvements. Our approach introduces the dynamic estimation of Spherical Harmonics (SH) bases using wavelets, which enhances the representation of dynamic objects in both space and time. Furthermore, the densification of point clouds across multiple frames contributes to faster convergence during model training. Extensive evaluations on the KITTI shows that DENSER outperforms state-of-the-art techniques by a substantial margin, showcasing its effectiveness in dynamic scene reconstruction. Future work will focus on extending this approach to deformable dynamic objects such as pedestrians and cyclists.

REFERENCES

- [1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Conference on robot learning*, pp. 1–16, PMLR, 2017.
- [2] LGSVL, “Lgsvl simulator.” <https://www.svlsimulator.com>. Accessed: 2024-08-03.
- [3] DeepDrive, “Deepdrive.” <https://github.com/deepdrive/deepdrive>. Accessed: 2024-08-03.
- [4] F. Mütsch, H. Gremmelmaier, N. Becker, D. Bogdolla, M. R. Zofka, and J. M. Zöllner, “From model-based to data-driven simulation: Challenges and trends in autonomous driving,” *arXiv preprint arXiv:2305.13960*, 2023.
- [5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” in *ECCV*, 2020.
- [6] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *TOG*, vol. 42, July 2023.
- [7] H. Dahmani, M. Bennehar, N. Piasco, L. Roldao, and D. Tsishkou, “Swag: Splatting in the wild images with appearance-conditioned gaussians,” *arXiv preprint arXiv:2403.10427*, 2024.
- [8] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” in *CVPR*, pp. 7210–7219, 2021.
- [9] J. Guo, N. Deng, X. Li, Y. Bai, B. Shi, C. Wang, C. Ding, D. Wang, and Y. Li, “Streetsurf: Extending multi-view implicit surface reconstruction to street views,” *arXiv preprint arXiv:2306.04988*, 2023.
- [10] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, “Block-nerf: Scalable large scene neural view synthesis,” in *CVPR*, 2022.
- [11] H. Turki, J. Y. Zhang, F. Ferroni, and D. Ramanan, “Suds: Scalable urban dynamic scenes,” in *CVPR*, 2023.
- [12] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone, and Y. Wang, “Emergenterf: Emergent spatial-temporal scene decomposition via self-supervision,” *arXiv preprint arXiv:2311.02077*, 2023.
- [13] T.-A.-Q. Nguyen, L. Roldão, N. Piasco, M. Bennehar, and D. Tsishkou, “Rodus: Robust decomposition of static and dynamic elements in urban scenes,” *arXiv preprint arXiv:2403.09419*, 2024.
- [14] Z. Wu, T. Liu, L. Luo, Z. Zhong, J. Chen, H. Xiao, C. Hou, H. Lou, Y. Chen, R. Yang, Y. Huang, X. Ye, Z. Yan, Y. Shi, Y. Liao, and H. Zhao, “Mars: An instance-aware, modular and realistic simulator for autonomous driving,” *CICAI*, 2023.
- [15] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, “Street gaussians for modeling dynamic urban scenes,” *arXiv preprint arXiv:2401.01339*, 2024.
- [16] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, “Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes,” in *CVPR*, pp. 21634–21643, 2024.
- [17] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, “Neural scene graphs for dynamic scenes,” in *CVPR*, pp. 2856–2865, 2021.
- [18] B. Attal, J.-B. Huang, C. Richardt, M. Zollhoefer, J. Kopf, M. O’Toole, and C. Kim, “HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling,” in *CVPR*, 2023.
- [19] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, “K-planes: Explicit radiance fields in space, time, and appearance,” in *CVPR*, 2023.
- [20] Z. Li, S. Niklaus, N. Snavely, and O. Wang, “Neural scene flow fields for space-time view synthesis of dynamic scenes,” in *CVPR*, 2021.
- [21] H. Lin, S. Peng, Z. Xu, T. Xie, X. He, H. Bao, and X. Zhou, “High-fidelity and real-time novel view synthesis for dynamic scenes,” in *SIGGRAPH Asia 2023 Conference Proceedings*, pp. 1–9, 2023.
- [22] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, “Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields,” *arXiv preprint arXiv:2106.13228*, 2021.
- [23] S. Peng, Y. Yan, Q. Shuai, H. Bao, and X. Zhou, “Representing volumetric videos as dynamic mlp maps,” in *CVPR*, pp. 4252–4262, 2023.
- [24] L. Song, A. Chen, Z. Li, Z. Chen, L. Chen, J. Yuan, Y. Xu, and A. Geiger, “Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields,” *TVCG*, vol. 29, no. 5, pp. 2732–2742, 2023.
- [25] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, “D-nerf: Neural radiance fields for dynamic scenes,” in *CVPR*, pp. 10318–10327, 2021.
- [26] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, “Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction,” *arXiv preprint arXiv:2309.13101*, 2023.
- [27] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, “Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction,” *arXiv preprint arXiv:2309.13101*, 2023.
- [28] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, “Unisim: A neural closed-loop sensor simulator,” in *CVPR*, 2023.
- [29] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, “Ewa volume splatting,” in *Proceedings Visualization, 2001. VIS’01*, pp. 29–538, IEEE, 2001.
- [30] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *CVPR*, 2012.
- [31] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, “Nerfstudio: A modular framework for neural radiance field development,” in *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH ’23, 2023.