Research on the Method of Applying 3D Gaussian Splatting Technology to Help Conduct Automatic Driving Training

Yanxin Wu*

School of Electric and Electronic Engineering, Shanghai University of Engineering Science, Shanghai, China

*Corresponding author: jackson5789123@gmail.com

Abstract. The purpose of this study is to explore the application of 3D Gaussian Splatting technology in automated driving training. In reality, the training of automatic driving models will produce great labor and economic consumption, but the use of virtual 3D scenes can significantly reduce this consumption, and at the same time have better training effects. Therefore, how to better use 3D for the training of automatic driving models will also be the future development direction of the field of automatic driving. In this paper, some major 3D Gaussian based dynamic scene modeling methods are summarized and some improvement schemes are proposed, including scene segmentation, new scene generation and dynamic scene rendering. This paper hopes to summarize and improve the current research, and finally propose that the generation target scene can be customized according to the training requirements of users' automatic driving models, so as to achieve more efficient and practical 3D scene generation.

Keywords: Automatic driving; 3D Gaussian; 3D Gaussian Splatting; 3D scene reconstruction technology.

1. Introduction

With the rapid development of artificial intelligence and its related fields, the technology of 3D scene generation has also made a breakthrough. Now 3D scene reconstruction technology has been able to handle more complex scenes and generate complete 3D structures from sparse input data. At present, 3D reconstruction technology has quite a number of applications, such as automatic driving, games and so on. By using the street picture to reconstruct the 3D scene and apply it to the training of the automatic driving model, instead of training in the real world, the human and economic cost of model training can be greatly reduced.

At present, to apply Neural Radiance Fields(NeRF), which is widely used in 3D scene reconstruction technology, to the modeling required by the training of automatic driving model, people are faced with many challenges, including: the training and reasoning time is long, which is difficult to meet the real-time application requirements; Under the condition of sparse viewing Angle, the reconstruction quality is insufficient, which is prone to detail loss and artifacts; The generalization ability is limited, requiring separate training for each new scene, and the adaptability to unseen scenes is poor; The modeling ability of high-resolution details and complex scenes (such as dynamic objects or large-scale scenes) is insufficient, and the problems of blurring and excessive storage requirements are prone to occur [1]. At the same time, since NeRF does not model geometry, it is unable to capture the association of objects' movements from frame to frame, and there is a shortage of dynamic modeling. These problems also lead to many problems when using NeRF to train the street scene required by the automatic driving model, especially the dynamic street scene. For example, its high computational cost makes it difficult for NeRF to meet the real-time application requirements.

However, 3D Gaussian Splatting proposes to introduce 3D Gaussian distribution as the representation of radiation field, which greatly optimizes the modeling and rendering efficiency of complex scenes, and provides a new and efficient solution for real-time new perspective generation[2].

Compared with NeRF, the training and rendering speed of the Gaussian Splatting technology is significantly accelerated, which can achieve high-quality real-time radiation field rendering. Meanwhile, because the 3D Gaussian Splatting technology adopts sparse point cloud as the initial

input, it also reduces the complexity of data acquisition. The adoption and improvement of these technologies is that it achieves a good balance between real-time and visual quality, which also makes it have a broad application prospect in the field of automatic driving.

With the introduction of 3D Gaussians Splatting technology, a large number of related researches based on this technology have also been published. Driving Gaussian and Street Gaussian, on the basis of applying 3D Gaussian to render the scene, propose the method of dividing the static background and dynamic foreground of the scene to realize the 4D scene modeling[3,4].

This paper summarizes some current researches on the application of 3D Gaussian Splatting technology in the direction of automatic driving model training, and proposes some new ideas and ideas based on these researches. These related methods and researches can be used as the reference and basis for the future implementation of the 3D Gaussian Splatting technology for automatic driving model training, and can also provide ideas and technical support for future research.

2. Methods and materials

2.1. 3D scene reconstruction

The first is the research on 3D point cloud. ParSeNet can be used to decompose 3D point cloud into parameterized surface fragments, including B-spline fragments as well as basic geometric primions. ParSeNet can deal with a richer class of geometric primions than the previous work, including basic primions such as plane, cylinder, cone, etc., with higher surface reconstruction accuracy[5].

Since NeRF technology, although excellent on small-scale scenes, is difficult to scale to large-scale (such as city-level) environments, Block-NeRF technology proposes a new method to partition large urban scenes into multiple independent small NeRF blocks for training and rendering to achieve the rendering of large scenes[6]. Thus solving the computational and memory bottlenecks of traditional NeRF in large scale scenes.

3D Gaussian Splatting proposes a new method to improve the efficiency and quality of rendering through 3D Gaussian representation[2]. It provides an efficient and differentiable solution for real-time radiation field rendering, which can achieve a good balance between training time and rendering quality. Compared with NeRF, the 3D Gaussian Splatting is more efficient in the processing of large scenes and complex geometric shapes, and is suitable for applications requiring fast response and high quality output.

With the proposal of 3D Gaussian Splatting technology, a large number of related and derivative researches begin to appear. Among them, the DreamGaussian framework applies the 3D Gaussian Splatting technology to the generation task, which significantly reduces the optimization time of 2D dimension raising method, improves the generation speed and maintains a high 3D content generation quality, making it more effectively deployed in practical applications[7]. Compared to the traditional NeRF method, this framework improves the generation efficiency by gradually increasing the density of 3D Gaussian. At the same time, an efficient algorithm is designed to extract the mesh from the 3D Gaussian, and the texture refinement is carried out in the UV space to enhance the generation quality.

In 3D scene reconstruction, Text-to-3D is also a very important research direction. DreamGaussian adopts a method based on the combination of 3D Gaussian Splatting with mesh extraction and UV spatial texture thinning to realize efficient 3D model generation[7]. COMPGS proposed 3D Gaussian initialization and dynamic SDS optimization methods based on 2D compositionality, which can generate multiple objects that can have complex interactions in 3D scenes[8]. However, COMPGS has some limitations in generating scenes containing background (such as ground, sky). COMPGS focuses on multi-object scene generation and interaction details, while DreamGaussian focuses more on the balance between generation speed and texture details.

2.2. Defination of 3D Gaussian

3D Gaussian is a way of using a Gaussian distribution to represent an object or point cloud in three-dimensional space, expressed as follows

$$\mathrm{G(x)} = \mathrm{e}^{-\frac{1}{2}(\mathrm{x})^{\mathrm{T}}\Sigma^{-1}(\mathrm{x})} \tag{1}$$

$\mathrm{G(x)}$ Is the 3D Gaussian value at the point, representing the covariance matrix, defined as follows:$x\Sigma$

$$\Sigma = RSS^{T}R^{T} \tag{2}$$

Where **R** is the rotation matrix, which describes the direction of the Gaussian distribution, and **S** is a diagonal matrix, which represents the scale of the Gaussian distribution along each spindle.

3D Gaussian represents the point of the scene obtained from the point cloud generated during camera calibration by a sparse 3D Gaussian distribution, while using adaptive optimization and density control, dynamically adjusting the number and parameters of 3D Gaussian to accurately represent the scene structure.

The rendering formula of 3D Gaussian is as follows:

$$\text{rendering: } \mathrm{C} = \sum_{\mathrm{i}\in\mathcal{N}} \mathrm{c_i}\alpha_\mathrm{i} \prod_{\mathrm{j}=1}^{\mathrm{i}-1}\left(1-\alpha_\mathrm{j}\right) \tag{3}$$

Where C represents the color of the final pixel, the color value of the i th Gaussian point, and the opacity of the i th Gaussian point.$c_i a_i$

The loss function is calculated by the following formula:

$$\mathrm{L} = (1-\lambda)\mathrm{L}_1(\mathrm{GT}-\mathrm{R}) + \lambda \mathrm{L}_{\text{D-SSIM}} \tag{4}$$

Where GT is the original image, R is the rendered image, and is the weight parameter used to control the balance between the L1L_1L1 λloss and the D-SSIM loss.

2.3. Dynamic 3D scene modeling

Both the Driving Gaussian and Street Gaussian frameworks can be used to efficiently and accurately reconstruct large-scale dynamic scenes in automatic driving[3,4]. Driving Gaussian decomposes the scene into static background and multiple dynamic objects by using the compound Gaussian projection method, and uses the incremental static Gaussian model and dynamic Gaussian graph to reconstruct and render the scene. Street Gaussians also decomposes the urban street scene into static background and dynamic foreground (such as moving vehicles), but adopts the 4D ball harmonic model to deal with the time change of color when modeling dynamic objects. By treating the tracked vehicle attitude as a learnable parameter, Street Gaussians enables the vehicle to show the appearance changing with time in the rendering process.

S3Gaussian proposes a self-supervised method to represent dynamic and static scenes through 4D Gaussian distribution, and use spatiotemporal field network to automatically decompose complex driving scenes[9]. The sparse point cloud generated by SfM is initialized, and the dynamic changes are captured by time coding and 4D spherical harmonic model, so that the dynamic foreground and static background are modeled as spatiotemporal and spatiotemporal dependent Gaussian distributions respectively. At the same time, adaptive density control strategy is adopted to increase Gaussian points in the high frequency region and reduce redundant points in the low frequency region, which significantly improves the representation efficiency and rendering quality. Finally, combined with the loss function, the self-supervised optimization of high-fidelity dynamic street view modeling and real-time rendering is realized.

Dynamic 3D Gaussian represents each object in the scene as a set of 3D Gaussian points that change over time[10].

Each Gaussian distribution has the following parameters:

1)The 3D center of each time step ($x_t, y_t z_t$)

2) a three-dimensional rotation of each time step, parameterized by a quaternion (qw_t, qx_t, qy_t, qz_t)

3) Three dimensional dimensions under standard deviation (sx, sy, sz)

4) Color (R, G, B)

5) Logarithm of opacity (consistent across all time steps) (o)

6) A background logarithm (consistent across all time steps) (bg)

The Dynamic Gaussian captures the dynamic changes of objects in the scene through the data of the time series. In each time step, the position, shape and color of the Gaussian points are adjusted according to the input sensor data. Then, through continuous optimization, the model minimizes the error between the current frame and the Gaussian point representation, while tracking the trajectory of the object, and dynamically reconstructing the object's motion, deformation and color change.

Through the framework of persistent dynamic view synthesis, the model is optimized to realize the tracking and reconstruction of every element in the scene, so as to solve the problems of 3D reconstruction of the dynamic scene, 6-DOF tracking and new perspective synthesis.

Compared with Street Gaussian and Driving Gaussian, Dynamic Gaussian, which can realize 6-DOF tracking without corresponding input and high frame rate rendering, pays more attention to persistent dynamic modeling and real-time processing capability. However, in the optimization process of Dynamic Gaussian, each Gaussian needs to be optimized, so the optimization time will be long. However, such high-precision modeling is not required in this study, so the Gaussian of an object can be regarded as a set for optimization, which is more efficient.

2.4. Simulation environment

There are many simulators available for autonomous driving training, such as AirSlim and CARLA, both of which are based on Unreal Engine, where AirSim can be integrated with actual autonomous driving hardware for real-time simulation and testing[11,12]. It also has a highly modular design that allows it to scale to support new vehicle types, hardware platforms and software protocols, as well as a variety of sensor models, all implemented in abstract interfaces that can be used independently. These features make it an ideal tool for researchers and developers to test algorithms in a simulation environment. CARLA focuses on the application of autonomous vehicles in urban environments, supports a large number of urban drivation-related scenarios, emphasizes urban traffic and vehicle behavior simulation, and provides a highly customizable platform for autonomous driving research, capable of testing and validating multiple driving strategies in a safe, controlled environment.

3. Discussion and analysis

At present, these existing methods are all based on the data collected by sensors and image data to generate the scene, only the scene that has been collected data can be generated without the ability to generate a new scene by itself. Even if it only needs to change part of the scene, it needs to resampling the scene in the case of a real scene. Will produce a lot of human and economic losses. Therefore, the above related methods are summarized and improved again in this paper, so as to realize the 3D scene reconstruction technology that can be customized and generated according to the training requirements of users for their automatic driving models. The following are some possible schemes that can be used to achieve these:

3.1. Scene Segmentation

According to the method mentioned in Street Gaussian, the point cloud of the generated scene is divided into static background and dynamic foreground, where $SS_b \in S\mathbf{S_o} \in \mathbf{SS_o} \supset \{\mathbf{O_1}, \mathbf{O_2}, \mathbf{O_3}, \ldots, \mathbf{O_N}\}$[4].For objects that do not change with time (such as buildings and roads), multiple frames of data can be accumulated to classify them as static objects. Foreground objects, such as vehicles and pedestrians, can be separated from the background by the motion path of the

object in space and classified as the foreground because of their significant displacement in multiple frames.

3.2. Scene Segmentation

After scene segmentation is completed, only its background part is extracted, and 2D image $S_b\mathbf{B}$ containing only the background can be rendered by eq2's rendering formula through the Gaussian of the background part. Where

$$B = \text{Render}(S_b) \tag{5}$$

Since Gaussian is optimized through two-dimensional images, generating Target in two-dimensional images, that is, other foreground objects (not foreground objects in the original scene), will also facilitate future Gaussian optimization, and generating Target in two-dimensional images is better than generating target in three-dimensional scenes in terms of computation and accuracy. $\widehat{\mathbf{O}}$,Therefore, add Target directly to the rendered image. The Target can be Mesh **M** of the Target from the database, and the vertex set of the Target Mesh is initialized as the center point of Gaussian. $\mathbf{V} \in \mathbf{M}$The Target can also generate Target Gaussian [7] through the Text-To-Image model Dream Gaussian according to the user's requirements, and then import it into the background to get a new scene.$\mathbf{S_b S'_{t_0}}$

3.3. Rendering of dynamic scene

In the previous part, this paper obtains the Gaussian set of the object in the new initial state. In order to conduct 4D modeling, the deformation of the real object in the data set is assigned. $\widehat{\mathbf{O}_{t_0}}$In driving Gaussian, because the deformation data of the object in each frame exists in the data set, and this deformation data is described by displacement vector, the modeling is directly based on dv. In our setting, there is no Dense Correspondence between the generated object and the real object, so there is no way to describe it directly by Displacement Vector. So an extra step of transformation is required. Given the state of the point cloud of a real object at two moments in the data set, since objects in the scene are rigid, $\mathbf{O_{t_1}}, \mathbf{O_{t_2}}$,And there is Dense Correspondence between the same object and, so the RT transformation matrix can be approximately calculated to represent the movement of the object between the two moments through the corresponding point displacements.$\mathbf{O_{t_1} O_{t_2}}$

$$O_{t_2} = RO_{t_1} + T, \tag{6}$$

Then pass $=\widehat{\mathbf{O}_{t_2}}\ \mathbf{R}\widehat{\mathbf{O}_{t_1}} + \mathbf{T}$. Then you can calculate the displacement between and. $\widehat{\mathbf{O}_{t_1}}\widehat{\mathbf{O}_{t_2}}\Delta\mathbf{s} \in R^{N\times 3}$It can then be interpolated according to demand to get the trajectory of its displacement, which is used to update the center of the Gaussian in each frame.

3.4. Optimization of the model

Since the background Gaussian has been optimized, in the process of 4D rendering, each frame is optimized. Given the camera pose, target mesh and 3D Gaussian are rendered with the same camera pose, and reconstruction loss is obtained with both. $\mathbf{S_o}$This reconstruction loss is used to optimize Gaussian.

$$\sum_{k=0}^{K} L_1\left(Render_m(M_k) - Render_g(O_k)\right), M_K \in M, O_k \in S_o \tag{7}$$

Which is based on the rendering formula of formula (3) to render Gaussian, $Render_m$ is differentiable mesh renderer [13].

4. Conclusion

At present, 3D Gaussian technology still needs to be improved in many areas. For example, in the process of large scene modeling, the number of Gaussian will increase rapidly during optimization,

resulting in excessive memory consumption. At present, some studies have proposed to limit the number of Gaussian generated to solve this problem, but how to achieve the best balance between the number of Gaussian and the appropriate optimization effect still needs to be further improved.

In this paper, the existing researches on 3D Gaussian Splatting, which can be applied to the training of automatic driving models, are summarized, and the existing problems and challenges of these researches are compared and expounded. Meanwhile, based on these researches, this paper also proposes some methods and ideas to improve and optimize the 3D Gaussian Splatting which can be used for automatic driving model training, hoping to provide certain theoretical basis and help in methods and ideas for subsequent research

However, the current improved scheme only proposes the replacement scheme based on the position and size of the original object. How to directly generate new objects in the scene without using the information of the original object is also the direction to be improved in the future. In addition, there is still a lot of exploration space in self-supervised training and how to allocate memory consumption more reasonably

References

[1] Gao R, Qi Y. A Brief Review on Differentiable Rendering: Recent Advances and Challenges. Electronics, 2024, 13(17): 3546.

[2] Kerbl B, Kopanas G, Leimkuhler T, et al. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Trans. Graph., 2023, 42(4): 139:1-139:14.

[3] Zhou X, Lin Z, Shan X, et al. DrivingGaussian: Composite Gaussian splatting for surrounding dynamic autonomous driving scenes//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 21634-21643.

[4] Yan Y, Lin H, Zhou C, et al. Street Gaussians for modeling dynamic urban scenes. arXiv preprint arXiv:2401.01339, 2024.

[5] Sharma G, Liu D, Maji S, et al. Parsenet: A parametric surface fitting network for 3d point clouds//Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. Springer International Publishing, 2020: 261-276.

[6] Tancik M, Casser V, Yan X, Pradhan S, Mildenhall B, Srinivasan P P, Barron J T, Kretzschmar H. Block-nerf: Scalable large scene neural view synthesis. In: CVPR, 2022.

[7] Tang J, Ren J, Zhou H, et al. DreamGaussian: Generative Gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653, 2023Ma Kunlong. Short term distributed load forecasting method based on big data. Changsha: Hunan University, 2014.

[8] Ge C, Xu C, Ji Y, et al. CompGS: Unleashing 2D Compositionality for Compositional Text-to-3D via Dynamically Optimizing 3D Gaussians. arXiv preprint arXiv:2410.20723, 2024.

[9] Huang N, Wei X, Zheng W, et al. $\textit {S}^ 3$ Gaussian: Self-Supervised Street Gaussians for Autonomous Driving. arXiv preprint arXiv:2405.20323, 2024.

[10] Luiten J, Kopanas G, Leibe B, et al. Dynamic 3d Gaussians: Tracking by persistent dynamic view synthesis. arXiv preprint arXiv:2308.09713, 2023.

[11] Shah S, Dey D, Lovett C, et al. Airsim: High-fidelity visual and physical simulation for autonomous vehicles//Field and Service Robotics: Results of the 11th International Conference. Springer International Publishing, 2018: 621-635.

[12] Dosovitskiy A, Ros G, Codevilla F, et al. CARLA: An open urban driving simulator//Conference on robot learning. PMLR, 2017: 1-16.

[13] Ravi N, Reizenstein J, Novotny D, et al. Accelerating 3d deep learning with pytorch3d. arXiv preprint arXiv:2007.08501, 2020.