BézierGS: Dynamic Urban Scene Reconstruction with Bézier Curve Gaussian Splatting

Supplementary Material

6. Implementation Details

6.1. Initialization

6.1.1. Bézier curve fitting

For points $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=0}^m$ that need to be initialized as a Bézier curve of degree n, we first reasonably initialize the Bézier parameters $\{t_i^0\}_{i=0}^m$ and solve for the control points' coordinates $\{\boldsymbol{p}_i \in \mathbb{R}^d\}_{i=0}^n$ using the least squares method. Then, we compute the closest Bézier parameter t_i^1 for each point $\boldsymbol{x}_i$ based on the obtained Bézier curve. This process is iteratively optimized until convergence. The workflow of our algorithm is presented in Algorithm 1.

Algorithm 1 Bézier initialization

Input: points $\{\boldsymbol{x}_i\}_{i=0}^m$ to be initialized as a Bézier curve
Output: control points $\{\boldsymbol{p}_i\}_{i=0}^n$ of Bézier curve

```
reasonably initialize the Bézier parameters {t_i^0}_{i=0}^m
k = 0
while not converge do
    solve {p_i^k}_{i=0}^n by {x_i}_{i=0}^m and {t_i^k}_{i=0}^m
    get {t_i^{k+1}}_{i=0}^m
    k = k + 1
end while
return {p_i^k}_{i=0}^n
```

Bézier parameters initialization. We leverages chord length parameterization to initialize the initial Bézier parameter $\{t_i^0\}_{i=0}^m$. Given points $\{\boldsymbol{x}_i\}_{i=0}^m$, we first compute the distances between consecutive points:

$$d_i = \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|, \quad i \in \{0, 1 \ldots, m-1\}. \tag{19}$$

The parameter $\{t_i^0\}_{i=0}^m$ are then assigned based on the cumulative chord length:

$$t_0^0 = 0,\ t_i^0 = \frac{\sum_{k=0}^{i-1} d_k}{\sum_{k=0}^{n-1} d_k}, \quad i \in \{1, 2 \ldots, n\}. \tag{20}$$

Control points fitting. Given the Bézier parameters $\{t_i^k\}_{i=0}^m$ and corresponding points $\{\boldsymbol{x}_i\}_{i=0}^m$, the control points $\{\boldsymbol{p}_i^k\}_{i=0}^n$ can be solved by minimizing the distance between the curve and the points. This can be formulated as a least squares problem:

$$\min ||\boldsymbol{B}\boldsymbol{P}^T - \boldsymbol{X}^T|| \tag{21}$$

where $\boldsymbol{B}_{ij} = b_{i,n}(t_j^k)$, $\boldsymbol{P} = \{\boldsymbol{p}_0^k, \ldots, \boldsymbol{p}_n^k\}$ and $\boldsymbol{X} = \{\boldsymbol{x}_0^k, \ldots, \boldsymbol{x}_m^k\}$. $b_{i,n}(t)$ is the Bernstein basis polynomial of degree n, which is defined as:

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i \in \{0, 1 \ldots, n\}. \tag{22}$$

Suppose that the matrix $\boldsymbol{B}$ has full column rank, then the solution for $\boldsymbol{P}$ can be expressed as $\boldsymbol{P} = \boldsymbol{X}\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{B})^{-1}$.

6.1.2. Trajectories initialization

For sequences with manual annotations, we use the annotated 3D bounding boxes to extract and merge LiDAR points from multiple frames of the same object, forming the initial point cloud of the object. The trajectory points $\{\boldsymbol{x}_i\}$ are obtained by transforming the merged LiDAR points into the world coordinate system, followed by Algorithm 1 to get the corresponding control points $\{\boldsymbol{p}_i\}$. For sequences without annotations, we employ a video instance tracking model [37] to extract the LiDAR points projected into the instance mask for each frame, and merge them as the initial point cloud of this instance. The center is computed by averaging the LiDAR point coordinates corresponding to each frame. Since our Bezier curve modeling can automatically correct errors, even this rough initialization leads to satisfactory reconstruction results.

6.1.3. Time-to-Bézier initialization

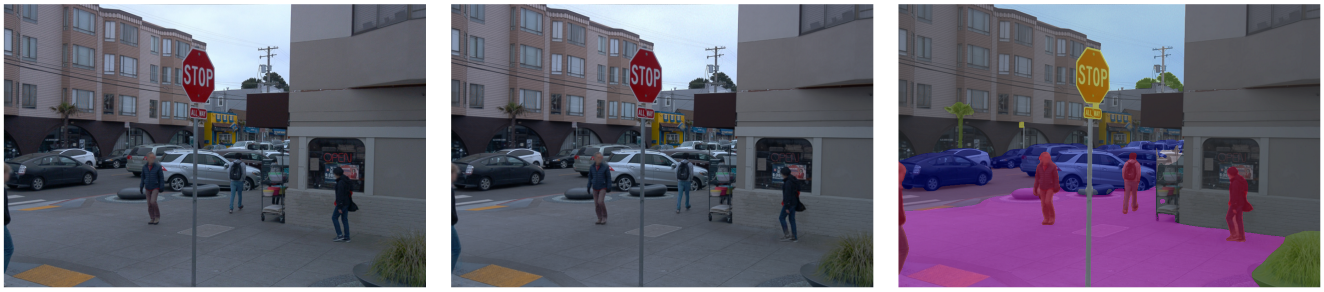
In the fitting process of Section 6.1.2, we simultaneously obtain the Bézier parameter $\{t_i\}$ corresponding to each point. By combining $\{t_i\}$ with the timestamp $\{\tau_i\}$, we obtain a new set of points $\{\boldsymbol{x}_i = (t_i, \tau_i)^T\}$ that need to be fitted. Similarly, we use Algorithm 1 to fit a Bézier curve and thus model the time-to-Bézier mapping.

6.2. Rotation

For rigid dynamics, the z-axis orientation remains fixed, and rotation occurs only within the xy-plane. Therefore, we rotate the Gaussian primitives based on the tangent direction of the trajectory projected onto the xy-plane rather than interpolate the quaternions. While this rotation modeling may not be ideal for non-rigid dynamics, experiments in the supplementary material show that it still effectively models the motions of non-rigid ones.

6.3. Hyperparameter Settings

The loss weights are tuned at a single sequence on Waymo [34] and applied directly to other sequences on Waymo and even nuPlan [19]. The consistently reliable reconstruction performance across different sequences and



(a) Ground truth (b) Render (c) Segmentation

Figure 6. **Reconstruction of complex motions.** Although the reconstructed pedestrian legs of *BézierGS* exhibit artifacts, the accurate semantic segmentation validates the correctness of visual perception for closed-loop simulation applications.

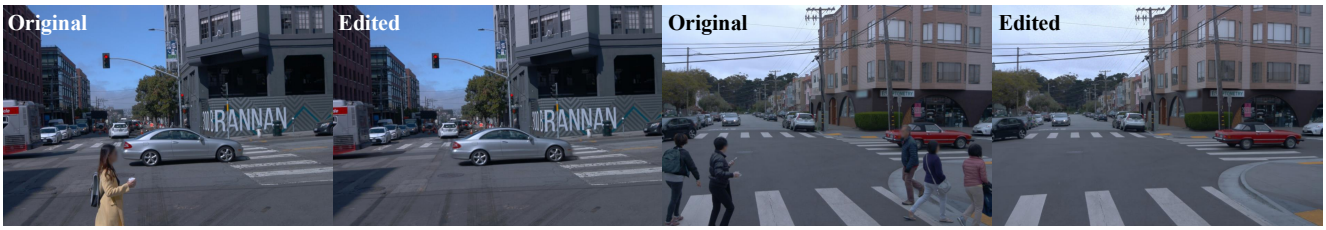


Figure 7. Remove the pedestrians.

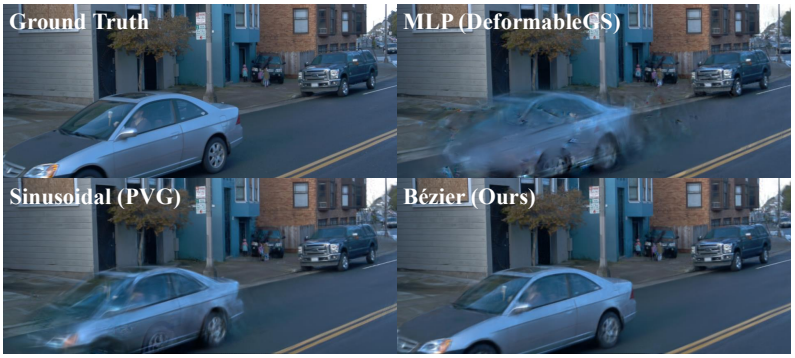


Figure 8. **Dynamic trajectory ablation** in novel view synthesis.

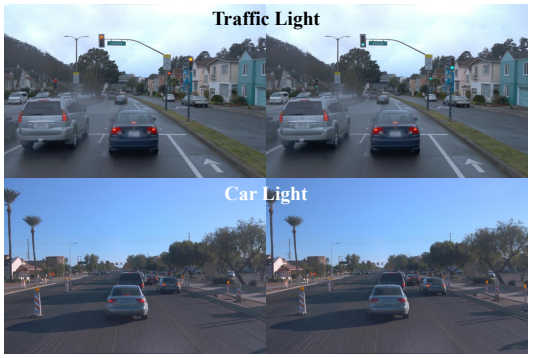


Figure 9. Model light variation using 4DSH.

datasets demonstrates the robustness of our method to hyperparameter settings, with no need for re-tuning.

7. Reconstruction of complex motions

Compared with fixed offset which can only model rigid objects, our Bézier offset is suitable for both rigid and non-rigid ones, achieving consistent and reliable reconstruction performance across diverse scenarios. For modeling the motion of deformable objects, the RGB loss plays a dominant role in capturing shape deformations, while the $\mathcal{L}_{icc}$ suppresses floaters. As illustrated in Figure 6 and the video **pedestrian.mp4**, our method achieves high reconstruction accuracy for the upper body of pedestrians, while certain artifacts may appear in the leg region. This limitation arises from the inherent smoothness of Bézier curves, which struggle to capture high-frequency, discontinuous motions such as leg swinging during walking. However, semantic segmentation [22] of the rendered images demonstrates that pedestrians are well segmented, ensuring the correctness of visual perception in closed-loop autonomous driving simulations. Furthermore, since our method and OmniRe [4] focus on different aspects, for applications requiring high-fidelity pedestrian reconstruction, we can integrate pedestrian module in OmniRe [4] to obtain improved results.

8. More results

Our method achieves high-quality reconstruction for both static and dynamic components, particularly demonstrating a significant advantage over state-of-the-art approaches in dynamic element reconstruction. By effectively capturing dynamic elements, our approach also excels in novel view synthesis at unseen timestamps, ensuring accurate and temporally consistent rendering. See more qualitative results in the supplementary video **BézierGS.mp4**.

Computational Cost. As shown in Table 3, our approach attains superior NVS quality while maintaining the training and rendering efficiency of 3DGS.

RTX A6000	GPU Mem. (GB)↓	Training Time (min)↓	FPS↑	PSNR ↑
DeformableGS [45]	22.7	123.1	4.9	29.52
HUGS [51]	8.8	61.4	46.1	29.34
Street Gaussians [41]	6.1	71.6	67.8	28.92
OmniRe [4]	12.4	103.5	58.3	29.41
PVG [1]	11.9	33.6	50.8	29.64
***BézierGS* (Ours)**	10.7	48.6	**88.7**	**31.51**

Table 3. **Computational Cost** on our Waymo sequences.

Accurate Separation. Our method enables effective separation of foreground and background components, including the removal of rigid and non-rigid objects. This further validates the robustness and generality of our approach in handling complex dynamic scenes (Figure 1(b) and Figure 7).

9. Limitations

We still have a few limitations for future improvement: First, our approach relies to some extent on the accuracy of the segmentation model, and low accuracy during training may lead to misinterpretations. Additionally, our method does not explicitly model lighting variations, which limits its ability to simulate complex lighting effects and shadow changes in dynamic environments. This challenge can be mitigated to some extent by employing the 4DSH approach introduced in Street Gaussians [41], as shown in Figure 9.