

4DSurf: High-Fidelity Dynamic Scene Surface Reconstruction

Supplementary Material

1. Derivation of SDF Flow from Motion of Gaussians

Let $\mathbf{x} \in \mathbb{R}^3$ denote a point in the canonical space of a Gaussian, with the origin at $(0, 0, 0)^\top$, the orientation of the Gaussian aligning with the axis of the canonical coordinate system, At timestep t , its position is

$$\mathbf{x}^t = \mathbf{R}^t \mathbf{x} + \mathbf{T}^t, \quad (1)$$

where $\mathbf{R}^t \in SO(3)$ and $\mathbf{T}^t \in \mathbb{R}^3$ are the rotation and translation of point \mathbf{x}^t at time t , respectively. While in the canonical space of the Gaussians' shape, the translation of a Gaussian is defined as $\boldsymbol{\mu}$. When transforming the Gaussian to the timestep t , its translation becomes $\boldsymbol{\mu} + \mathbf{v}^t t$ according to the definition in Sec. 4.1 "Gaussian Velocity Field", where $\boldsymbol{\mu}$ represents the Gaussian center in the canonical space of the shape and \mathbf{v}^t means the line velocity of the Gaussian at timestep t . Therefore, we can use the motion of Gaussians to represent the motion of points and define $\mathbf{T}^t = \boldsymbol{\mu} + \mathbf{v}^t t$. At a later timestep $t + \Delta t$, the updated position becomes

$$\mathbf{x}^{t+\Delta t} = \mathbf{R}^{t+\Delta t} \mathbf{x} + \mathbf{T}^{t+\Delta t}, \quad (2)$$

where $\mathbf{R}^{t+\Delta t} \in SO(3)$ and $\mathbf{T}^{t+\Delta t} \in \mathbb{R}^3$ are the rotation and translation of point $\mathbf{x}^{t+\Delta t}$ at time $t + \Delta t$, respectively. Note that $\mathbf{T}^{t+\Delta t} = \boldsymbol{\mu} + \mathbf{v}^{t+\Delta t} (t + \Delta t)$, where $\mathbf{v}^{t+\Delta t}$ means the velocity of the point at timestep $t + \Delta t$.

Relative Motion. The relative rotation \mathbf{R}_r between the two timesteps is

$$\mathbf{R}_r = \mathbf{R}^{t+\Delta t} (\mathbf{R}^t)^{-1}. \quad (3)$$

The translation offset $\Delta \mathbf{T}$ between the two timesteps is

$$\Delta \mathbf{T} = \mathbf{T}^{t+\Delta t} - \mathbf{T}^t \quad (4)$$

$$= \boldsymbol{\mu} + \mathbf{v}^{t+\Delta t} (t + \Delta t) - \boldsymbol{\mu} - \mathbf{v}^t t \quad (5)$$

$$= (\mathbf{v}^{t+\Delta t} - \mathbf{v}^t) t + \mathbf{v}^{t+\Delta t} \Delta t. \quad (6)$$

Taking the limit under the assumption of constant velocity, we have: $\lim_{\Delta t \rightarrow 0} \frac{\Delta \mathbf{T}}{\Delta t} = \mathbf{v}$. Hence, the relative motion can be written as

$$\mathbf{x}^{t+\Delta t} - \mathbf{x}^t = (\mathbf{R}^{t+\Delta t} - \mathbf{R}^t) \mathbf{x} + (\mathbf{T}^{t+\Delta t} - \mathbf{T}^t) \quad (7)$$

$$= (\mathbf{R}_r \mathbf{R}^t - \mathbf{R}^t) \mathbf{x} + \Delta \mathbf{T} \quad (8)$$

$$\approx \Delta \mathbf{R} \mathbf{R}^t \mathbf{x} + \Delta \mathbf{T}, \quad (9)$$

where $\Delta \mathbf{R} \approx \mathbf{R}_r - \mathbf{I}$ according to Rodrigues' rotation formula.

Scene Flow. Taking the limit as $\Delta t \rightarrow 0$, we obtain the scene flow:

$$\frac{\partial \mathbf{x}}{\partial t} = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{x}^{t+\Delta t} - \mathbf{x}^t}{\Delta t} \quad (10)$$

$$\approx \lim_{\Delta t \rightarrow 0} \frac{\Delta \mathbf{R} \mathbf{R}^t \mathbf{x} + \Delta \mathbf{T}}{\Delta t} \quad (11)$$

$$= \boldsymbol{\omega} \times \mathbf{R}^t \mathbf{x} + \mathbf{v}, \quad (12)$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ denote the angular and line velocities of the point on the Gaussian disk, respectively. According to the above formula, we can explicitly model the scene flow of the Gaussians by using our Gaussian Velocity Fields.

SDF Flow. Similar to [8], the temporal derivative \mathbf{f} of the SDF is:

$$\mathbf{f} = \frac{\partial s}{\partial t} = \lim_{\Delta t \rightarrow 0} \frac{\Delta s}{\Delta t} \quad (13)$$

$$= - \left(\frac{\partial \mathbf{x}}{\partial t} \right)^\top \mathbf{n}(\mathbf{R}^t \mathbf{x}) \quad (14)$$

$$= - (\boldsymbol{\omega} \times \mathbf{R}^t \mathbf{x} + \mathbf{v})^\top \mathbf{n}(\mathbf{R}^t \mathbf{x}), \quad (15)$$

where $\mathbf{n}(\mathbf{R}^t \mathbf{x}) \in \mathbb{R}^3$ is the surface normal at point $\mathbf{R}^t \mathbf{x}$. It should be noted that in order to simplify the computation, we use the center of Gaussian to compute the SDF flow.

2. SDF Flow from Geometry Changes

Let us start from Sec. 4.1 "SDF Flow from Geometry Changes" in the main paper. We approximate the SDF values using differences along the optical axis, following prior work [2] which has shown that this approximation is sufficiently accurate. Notably, KinectFusion [9] also reports that the results show no considerable difference whether the SDF is computed along the viewing ray or the optical axis. As a result, here is the final derivation of the SDF flow $\tilde{\mathbf{f}}$ from geometry changes:

$$\tilde{\mathbf{f}}_i^t = \frac{\partial \tilde{s}(\boldsymbol{\mu}_i^t, t)}{\partial t} \quad (16)$$

$$= \frac{\partial \hat{D}(\mathbf{p}^*, t)}{\partial t} - \frac{\partial d(\boldsymbol{\mu}_i^t, t)}{\partial t}, \quad (17)$$

where, $\tilde{s}(\boldsymbol{\mu}_i^t, t)$ denotes the approximated SDF values for Gaussian center $\boldsymbol{\mu}_i^t$, $d(\boldsymbol{\mu}_i^t, t) \in \mathbb{R}$ denotes the distance from the camera origin to the $\boldsymbol{\mu}_i^t$ along the optical axis, and $\hat{D}(\mathbf{p}^*, t)$ represents the corresponding surface depth point at the projected pixel \mathbf{p}^* on the depth map. In our implementation, we compute $\frac{\partial \hat{D}(\mathbf{p}^*, t)}{\partial t}$ and $\frac{\partial d(\boldsymbol{\mu}_i^t, t)}{\partial t}$ by using

central finite differences with respect to time for the convenience of computation, since it is complicated to get the analytical gradient of both of them.

3. More Details of Optimization

We add more details of our optimization here, as the supplementary of Sec. 4.4 “Optimization” in the main paper. The optimization process of our method consists of a photometric loss, three regularization losses, and a mask loss.

Photometric Loss: The photometric loss \mathcal{L}_{img} is responsible for image reconstruction and follows previous works [4, 17]. Specifically, we adopt a combination of an \mathcal{L}_1 loss and a D-SSIM term: $\mathcal{L}_{\text{img}} = (1 - \lambda)\mathcal{L}_1(\mathbf{I}, \mathbf{I}^*) + \lambda\mathcal{L}_{\text{D-SSIM}}(\mathbf{I}, \mathbf{I}^*)$, where, $\lambda = 0.2$ is the hyper-parameter, \mathbf{I} and \mathbf{I}^* denote the ground-truth and rendered images, respectively.

Normal Alignment & Depth Distortion Regularization:

Two regularization losses are introduced from 2DGS [3] are also included. The first is the normal alignment regularization, which aligns the estimated normal of each Gaussian with the normal derived from the rendered depth: $\mathcal{L}_n = \sum_i w_i (1 - \mathbf{n}_i^\top \mathbf{N})$, where w_i denotes the blending weight at the intersection point, \mathbf{n}_i is the normal of the Gaussian facing the camera, and \mathbf{N} is the pseudo normal derived from the depth map. The second is the depth distortion regularization, which enforces local depth consistency across intersected Gaussians: $\mathcal{L}_d = \sum_{i,j} w_i w_j |z_i - z_j|$, where z_i is the intersected depth value.

SDF Flow Regularization: In addition, we introduce an SDF flow regularization $\mathcal{L}_{\text{flow}}$, which encourages temporally consistent surface evolution by matching SDF flow from the motion of Gaussians and geometry changes: $\mathcal{L}_{\text{flow}} = \sum_i |\mathbf{f}_i^t - \tilde{\mathbf{f}}_i^t|$, where \mathbf{f}_i^t and $\tilde{\mathbf{f}}_i^t$ denote the SDF flow from the motion of Gaussians and geometry changes of the i^{th} Gaussian at timestep t .

Total Training Objective: Finally, following prior works [8, 12], we also incorporate a mask loss to reduce background artifacts. Formally, $\mathcal{L}_m = \mathcal{L}_1(\mathbf{M}, \mathbf{M}^*)$, where \mathbf{M} and \mathbf{M}^* denote the ground-truth and rendered alpha masks. Then the total training objective is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{img}} + \lambda_1 \mathcal{L}_n + \lambda_2 \mathcal{L}_d + \lambda_3 \mathcal{L}_{\text{flow}} + \lambda_4 \mathcal{L}_m, \quad (18)$$

where $\{\lambda_i\}$ are balancing hyper-parameters.

4. More Details of Experimental Settings

More Details of Datasets. From the CMU Panoptic dataset [5], we use four scenes: Ian3, Haggling b2, Band1, and Pizzal, each captured with a circular rig of 10 RGB-D cameras at 1920×1080 resolution. Each scene spans 24 timesteps and provides ground-truth point clouds. From the Hi4D dataset [16], we use

six scenes: Backhug02, Basketall13, Fight17, Football13, Talk22, and Cheers37, each captured with 8 surrounding RGB cameras at 940×1280 resolution. On average, each sequence contains 118 timesteps and each timestep is annotated with a high-quality textured 3D mesh. Compared with CMU Panoptic, Hi4D features more complex motions, longer sequences, and frequent human-human interactions.

Additional Implementation Details. We use \mathcal{L}_n and \mathcal{L}_d after 3000 iterations. When the model training is roughly stable, at 8000 iterations, we enable $\mathcal{L}_{\text{flow}}$. The weights λ_{1-4} are set to 0.08, 1000, 0.1, and 0.1, respectively. We adopt opacity cull and densify operation and set opacity cull thread to 0.01. The network structure of Gaussian Velocity Fields follow the deformation network structure in [15], comprising 8 linear layers with hidden dimension 256, and three heads that are used to predict velocity, angular velocity, and expansion velocity. We model the Gaussian Velocity Fields following [17], with the initial learning rate of 1.6×10^{-4} , which is decayed to 1.6×10^{-6} at the end of training. If we enable incremental motion tuning, the initial learning rate is set to 1.6×10^{-5} . Note that SH coefficients of each Gaussian are not time-dependent (but temporal appearance variations can be induced by geometric motion and view-dependent effects) and we do not follow the graph of control points in Dynamic 2DGS [17].

Additional Description of Some Selected Baselines.

About GauSTAR [18], we use its publicly available code for training, but to be fair, we removed the depth prior dependency. NeRF-based methods [1, 8, 11] are omitted from the Hi4D comparison as they require extremely long training time on such long sequences; Space-Time-2DGS [12] is excluded due to unavailable code, with results reported only on the CMU Panoptic. We also exclude MonoFusion [13], DG-Mesh [6], and MaGS [7] due to their reliance on priors from foundation models or task-specific assumptions. MonoFusion incorporates priors from multiple foundation models but ours does not require such priors, which means direct comparison is not fair. DG-Mesh relies on DPSPR [10] which is designed for single-object watertight surface and it fails in our multi-shapes settings. MaGS, tailored for monocular videos on single objects, refines only coarse meshes with few vertices and cannot handle multiple shapes (said in their limitation), making it unsuitable for our settings.

Description of Chamfer Distance. We evaluate our method with *Accuracy* (Acc), *Completeness* (Comp), and *Overall distance* (Overall) of Chamfer distance. Specifically, given the ground-truth point cloud \bar{P} and the predicted point cloud P , the Acc and Comp are defined as:

$$\text{Acc} = \frac{1}{|P|} \sum_{p \in P} \min_{\bar{p} \in \bar{P}} \|p - \bar{p}\|_2, \quad (19)$$

Table 1. The impact of different segment sizes.

Segment Size	Acc ↓	Comp ↓	Overall ↓
3	0.94	0.46	0.70
5	0.89	0.45	0.67
8	1.12	0.68	0.90

$$\text{Comp} = \frac{1}{|P|} \sum_{\bar{p} \in P} \min_{p \in P} \|p - \bar{p}\|_2. \quad (20)$$

The Overall (Chamfer Distance) is then defined as the average of the Acc and Comp.

5. Limitations and Future Work

Our current method assumes access to foreground masks. While this requirement is shared by many existing reconstruction and novel-view synthesis approaches [6, 14, 17]. Future work can explore how to remove the reliance on masks. In addition, although our approximated SDF values already achieve high accuracy in practice, there is still room for further refinement. Possible methods to be used in the future include estimating SDF values from multiple viewpoints and taking the average as the final SDF value.

6. Analysis on Segment Size

We investigate the effect of different segment sizes on our method without IMT and show results in Tab. 1. We evaluate our method with three segment sizes of 3, 5, and 8, each accompanied by an additional virtual timestep to enhance temporal continuity. Among these settings, a segment size of 5 yields the best overall performance.

7. More Qualitative Results

To further validate the robustness and generalization ability of our method, on the basis of Sec. 5.2 “Comparisons” from the main paper, we provide additional qualitative reconstructions from both the CMU Panoptic [5] and Hi4D datasets [16]. These examples cover a wide range of motion dynamics, scene complexity, and human interactions.

7.1. For CMU Panoptic Dataset

Figure 1 show results of scene Haggling b2 of the CMU Panoptic dataset. Our method accurately reconstructs multi-instance motions and preserves consistent geometry throughout temporal sequences.

7.2. For Hi4D Dataset

We show more examples on the Hi4D dataset in Figs. 2 to 5. The results demonstrate that our method can handle a variety of human activities, including the highly dynamic scene Football118 in Fig. 2 and the more subtle or close-contact motions Talk22, Cheers37, and Backhug02

Table 2. Comparison between Visual Hull and our method. For the CMU Panoptic dataset [5], our CD is in mm; for the Hi4D dataset [16], our CD is in cm.

	CMU Panoptic [5]	Hi4D [16]
Visual Hull	24.2	11.9
Ours wo IMT-64	11.5	0.67

in Figs. 3 to 5, achieving high-quality geometry and smooth temporal transitions. All six scenes involve close human interactions with large deformations and topological changes. Note that scene Backhug02 in Fig. 5 shows two separate bodies gradually coming into contact and forming large merged surfaces. It further demonstrates our method’s robustness to handle such challenges of large deformations.

7.3. Video Comparisons

We also provide some video comparisons of our method and baselines in the Video Supplementary Material. Please check the video file: “Video-Supp-720.mp4”.

8. Visual Hull Visualization and Comparisons

We show an example of initialized visual hull of the scene Band1 from all views of the 1st timestep. From Fig. 6, we can see that visual hull can be treated as a very well initialization. In addition, to verify that our model does not simply converge to the visual hull, we construct a polygonal visual hull based on the masks of all frames at each time step as a baseline for comparison. We validated our findings on two datasets, as shown in Tab. 2. Its inferior Chamfer distances confirm that our method does not simply converge to a visual hull.

9. Temporal Stability Visualization

As described in Sec. 5.3 “Temporal Stability” in the main paper, our method achieves the lowest standard deviation. In Fig. 7, we take one scene Backhug02 as an example to visualize its Acc, Comp and Overall across all timesteps, and compared with other dynamic surface reconstruction methods (Sparse2DGS [14], Dynamic-2DGS [17], GauS-TAR [18]). It is not difficult to find that three metrics of our methods are the most stable across all timesteps.

References

- [1] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *Advances in Neural Information Processing Systems*, 35:967–981, 2022. 2
- [2] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *CVPR*, 2024. 1

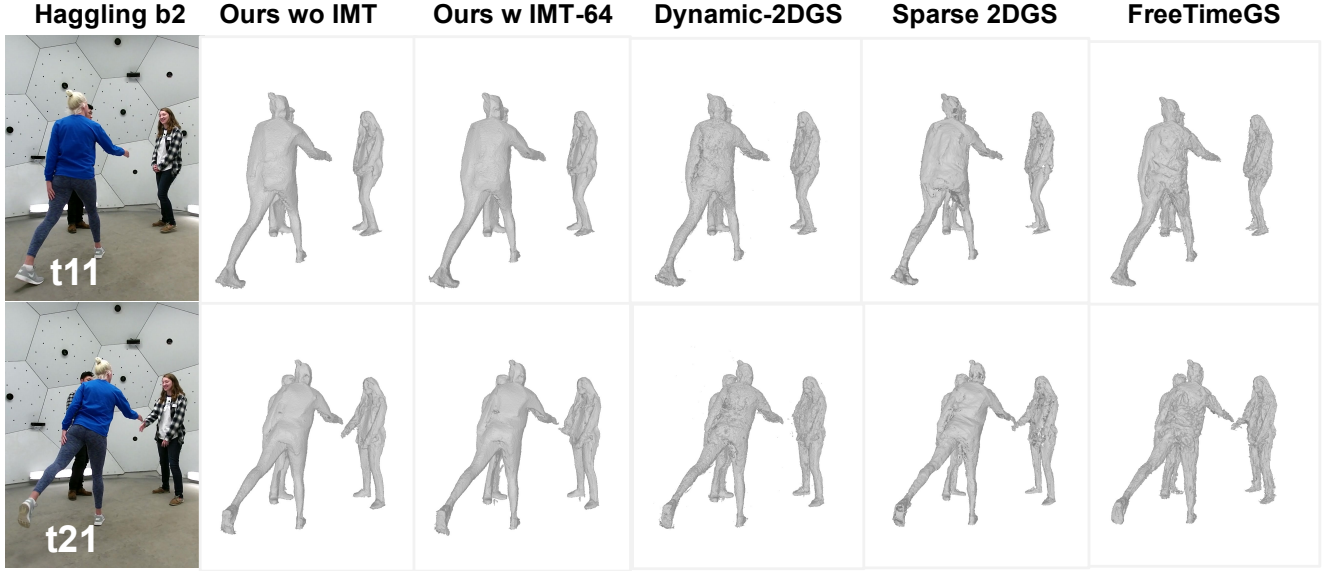


Figure 1. Haggling-b2 scene.

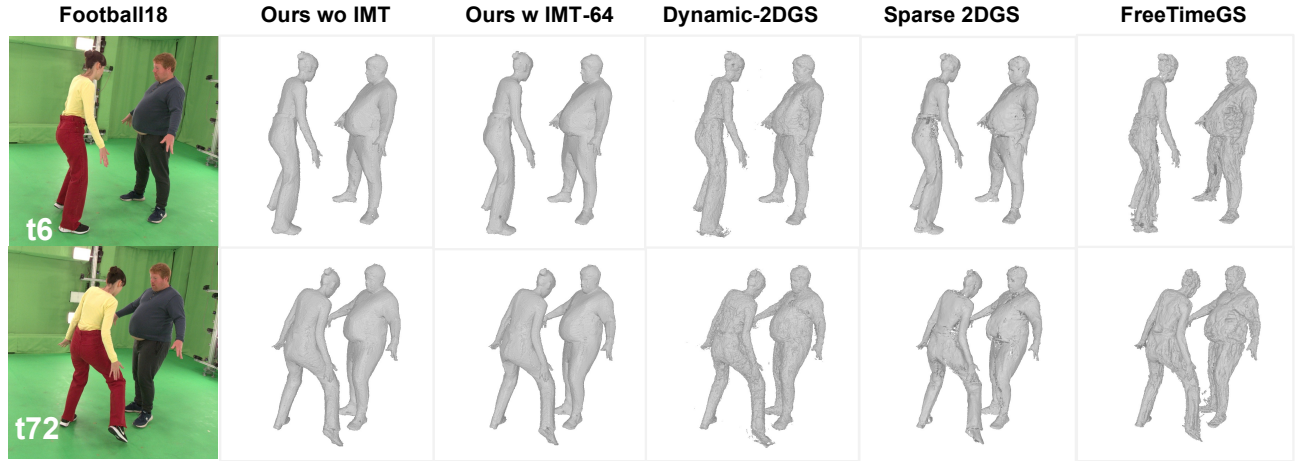


Figure 2. Football18 scene.

- [3] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 2
- [4] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4220–4230, 2024. 2
- [5] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pages 3334–3342, 2015. 2, 3
- [6] Isabella Liu, Hao Su, and Xiaolong Wang. Dynamic gaussians mesh: Consistent mesh reconstruction from dynamic scenes. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3
- [7] Shaojie Ma, Yawei Luo, Wei Yang, and Yi Yang. Mags: Reconstructing and simulating dynamic 3d objects with mesh-adsorbed gaussian splatting. *arXiv preprint arXiv:2406.01593*, 2024. 2
- [8] Wei Mao, Richard Hartley, Mathieu Salzmann, et al. Neural sdf flow for 3d reconstruction of dynamic scenes. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2
- [9] Richard A Newcombe, Shahram Izadi, Otmar Hilliges,

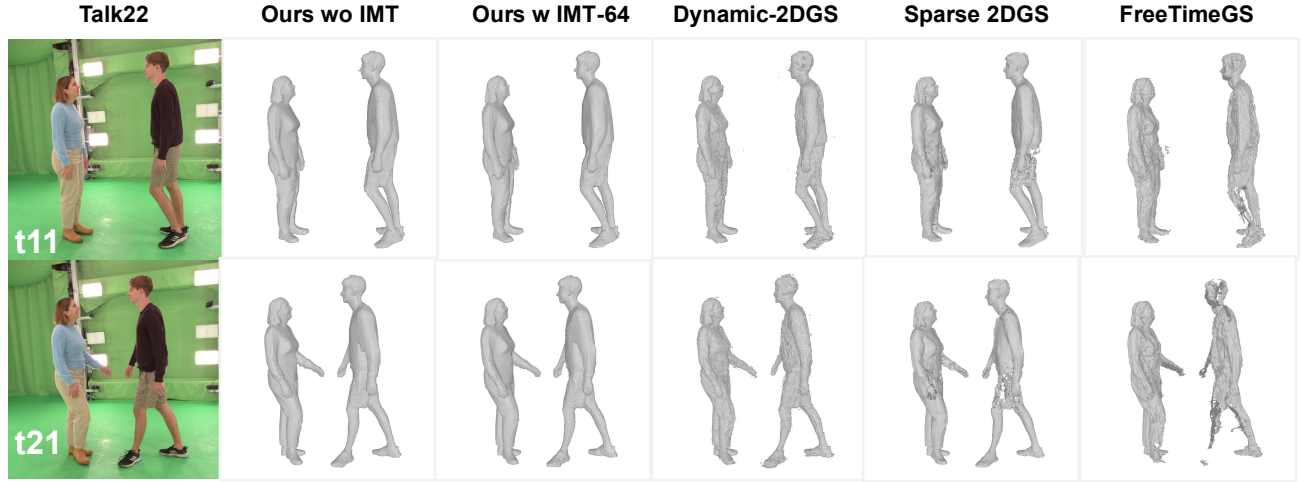


Figure 3. Talk22 scene.

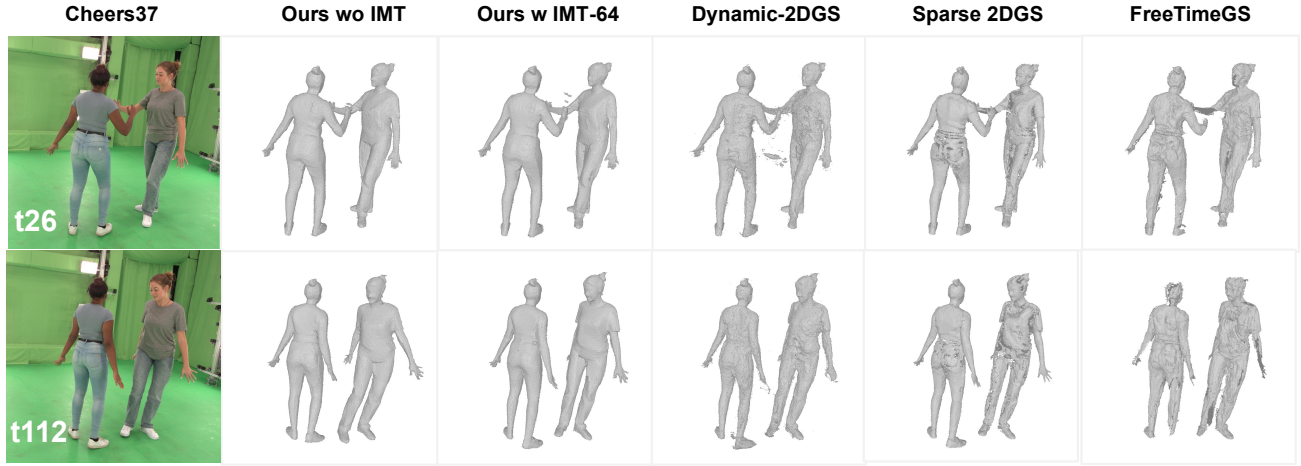


Figure 4. Cheers37 scene.

- David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 1
- [10] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems*, 34:13032–13044, 2021. 2
- [11] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2023. 2
- [12] Shuo Wang, Binbin Huang, Ruoyu Wang, and Shenghua Gao. Space-time 2d gaussian splatting for accurate surface reconstruction under complex dynamic scenes. *arXiv preprint arXiv:2409.18852*, 2024. 2
- [13] Zihan Wang, Jeff Tan, Tarasha Khurana, Neehar Peri, and Deva Ramanan. Monofusion: Sparse-view 4d reconstruction via monocular fusion. *arXiv preprint arXiv:2507.23782*, 2025. 2
- [14] Jiang Wu, Rui Li, Yu Zhu, Rong Guo, Jinqiu Sun, and Yan-ning Zhang. Sparse2dgs: Geometry-prioritized gaussian splatting for surface reconstruction from sparse views. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11307–11316, 2025. 3
- [15] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20331–20341, 2024. 2
- [16] Yifei Yin, Chen Guo, Manuel Kaufmann, Juan Zarate, Jie Song, and Otmar Hilliges. Hi4d: 4d instance segmentation

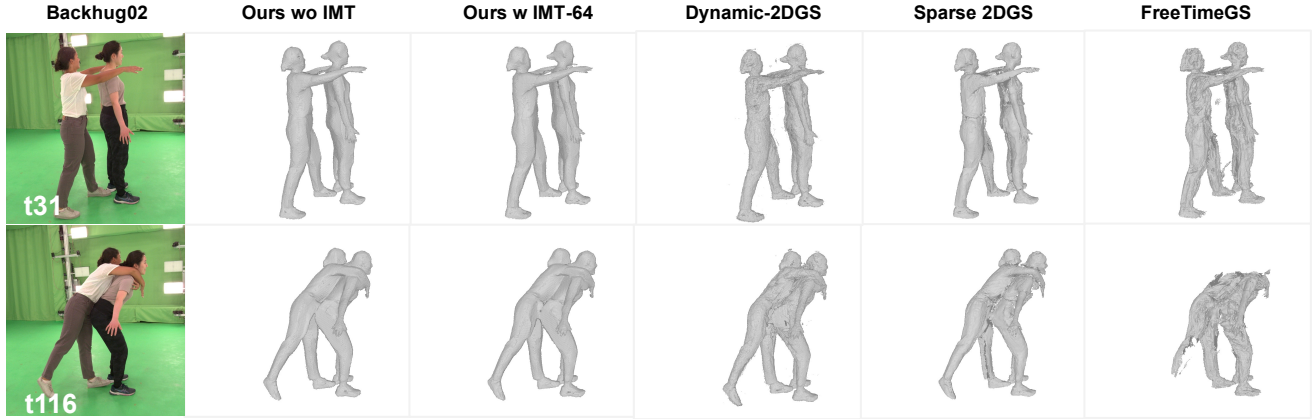


Figure 5. Backhug02 scene.



Figure 6. Visual hull of Band1 scene, which is constructed from all views of the 1st timestep.

of close human interaction. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

- [17] Shuai Zhang, GuanJun Wu, Zhoufeng Xie, Xinggang Wang, Bin Feng, and Wenyu Liu. Dynamic 2d gaussians: Geometrically accurate radiance fields for dynamic objects. *arXiv preprint arXiv:2409.14072*, 2024. 2, 3
- [18] Chengwei Zheng, Lixin Xue, Juan Zarate, and Jie Song. Gaustar: Gaussian surface tracking and reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16543–16553, 2025. 2, 3

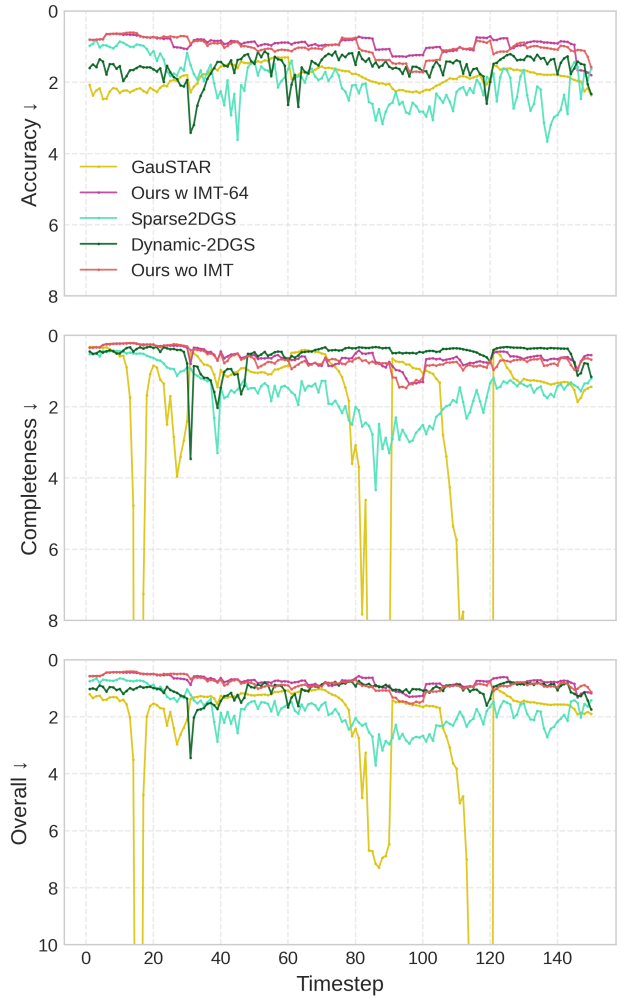


Figure 7. Temporal stability of Acc, Comp and Overall compared with other baselines across all timesteps for scene Backhug02.