Minute4D: Training High-Fidelity 4D Gaussian Splatting in One Minute

Binjian Xie[1,2*], Chenhui Shi[1*], Pengju Zhang[1†], and Yihong Wu[1,2,3†]

[1] State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] YUKUN Intelligent World, Beijing, China
{xiebinjian2022, shichenhui2022, pengju.zhang}@ia.ac.cn, yhwu@nlpr.ia.ac.cn

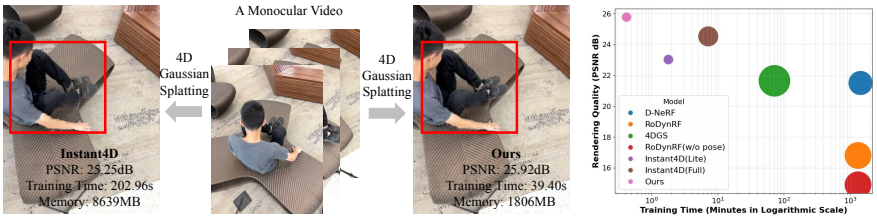

Fig. 1: (a) Minute4D reconstructs a 4D Gaussian scene from an unconstrained monocular video, achieving higher rendering quality along with faster training and lower GPU memory consumption. (b) Bubble chart compares Minute4D with recent methods, where a smaller bubble size indicates lower GPU memory consumption.

Abstract. Dynamic view synthesis has seen significant advances, yet reconstructing scenes from uncalibrated, casual videos remains challenging due to ambiguity between camera and object motion, inadequate view coverage and slow optimization. In this work, we present **Minute4D**, a novel and efficient approach for high-fidelity 4D scene reconstruction from monocular videos using Gaussian Splatting. Our Gaussian initialization begins with an efficient geometric recovery leveraging pre-trained visual foundation models. To improve reconstruction fidelity, we introduce a Segmentation-Tracking Enhancement module that leverages 2D semantic priors and 3D point tracking to jointly enhance the geometric accuracy and completeness of the initial Gaussians. To reduce redundancy and accelerate Gaussian optimization, we propose a Loss-Guided Density Control strategy that adaptively densifies and prunes Gaussians based on multi-view photometric loss. Our method reduces the optimization time to approximately 40 seconds for a typical 400-frame video, yielding a speed-up of at least **5**×. Extensive experiments show that our method not only enables substantially faster optimization but also achieves superior performance across several benchmarks.

*: Equal contribution.
†: Co-corresponding authors.

Keywords: 4D Reconstruction · Dynamic View Synthesis · Gaussian Splatting · Training Acceleration

1 Introduction

Reconstructing 4D scenes from uncalibrated, casual videos is a fundamental challenge in computer vision, crucial for applications such as robotics, autonomous driving, virtual reality and augmented reality. Although advances in Neural Radiance Field (NeRF) [28] and 3D Gaussian Splatting (3DGS) [18] have revolutionized static 3D reconstruction, adapting these techniques to capture dynamic scenes, especially using only monocular videos, still remains challenging. Object motion and occlusion increase scene complexity, and irregular camera trajectories further hinder efficient and accurate reconstruction.

Several methods have attempted to tackle this challenge, with the development of visual foundation models. RoDynRF [24] uses a pre-trained instance segmentation model [11] to separate dynamic objects from the static background and then jointly optimizes camera parameters and radiance fields during training. RoDyGS [16] uses a geometry foundation model [22] to estimate camera parameters and employs an object tracking model [47] to obtain dynamic masks. MoSca [21] leverages flow priors from a 2D optical flow model [39] to estimate dynamic masks, while using monocular depth estimation models [13,14,33] and 2D point tracking models [6,17,43] to obtain a global point cloud. Nevertheless, these methods above require hours to reconstruct even a short, casual video. Recently, Instant4D [26] jointly estimates camera parameters and a global point cloud using a deep visual SLAM system [23] and applies a grid pruning strategy to reduce Gaussian redundancy during initialization. However, reconstructing a scene of hundreds of frames still takes several minutes. In addition, Instant4D initializes 4D Gaussians based sorely on per-frame depth maps, which ignores correspondences of dynamic points across frames. Consequently, frame-wise overfitting of Gaussians occurs, where they adapt to regions visible in individual frames at the expense of global consistency, harming reconstruction quality.

To address the issues above, we propose a novel dynamic scene reconstruction method named Minute4D. Minute4D takes unconstrained casual videos as input and is able to reconstruct 4D Gaussians with high quality and high efficiency, as shown in Fig. 1. Our method is based on two key ideas. First, we design a Segmentation-Tracking Enhancement module, which aims to precisely separate dynamic foreground from static background and fill occluded regions of dynamic objects, thereby improving reconstruction quality. In particular, we first employ MegaSAM [23] to obtain initial dynamic masks, which are then refined by leveraging the semantic prior of SAM2 [36]. Guided by these refined masks, we use a 3D point tracking model TAPIP3D [51] to estimate the 3D trajectories of dynamic points. These trajectories enable the completion of regions that are occluded in some frames but visible in others, thus enhancing the geometric accuracy and completeness of the initial Gaussians. Second, we propose a novel Loss-Guided Density Control strategy that helps reduce Gaussian

redundancy and substantially improve optimization efficiency. Specifically, during optimization, we compute per-frame error maps by comparing the rendered images with their ground-truth counterparts. Based on these maps and the temporal properties of each Gaussian, we derive two time-weighted scores: one to guide densification and the other to guide pruning. Our main contributions are summarized as follows:

- We propose a Segmentation-Tracking Enhancement module to accurately separate dynamic objects from the static background and to effectively fill regions where dynamic objects are occluded in some frames but visible in others, thereby enhancing reconstruction completeness and improving the accuracy of novel view synthesis.
- We propose a Loss-Guided Density Control strategy, which uses multi-view error information during optimization to effectively reduce the redundant Gaussians, substantially improving the optimization efficiency.
- Experimental results demonstrate that the proposed method achieves superior reconstruction quality and the fastest optimization speed across several benchmarks, with a speed-up of at least 5× compared to other advanced methods.

2 Related Work

2.1 Dynamic Scene Reconstruction for Novel View Synthesis

Dynamic scene reconstruction is highly practical for real-world applications and has been studied for decades. Earlier methods, such as [1, 2, 30], employed discrete point clouds or meshes as scene representations. These representations lack smoothness, thus limiting the quality of novel view synthesis. In recent years, NeRF [28] has seen widespread development in 3D reconstruction due to its differentiable scene representation and simple structure. Building upon NeRF, methods like [3,25,34] have successfully modeled dynamic scenes, achieving ground-breaking novel view synthesis through volumetric rendering. However, due to the implicit nature of NeRF, these methods suffer from very slow rendering speeds. To overcome this limitation, 3DGS [18] has emerged as an alternative mainstream approach, leveraging its explicit continuous representation and rapid differentiable rasterization. To extend 3DGS to dynamic scenes, one line of work employs MLPs or low-rank K-planes to build deformation fields for Gaussian motion [19, 42, 44, 49]. Although expressive, the added complexity of learning continuous deformation fields tends to slow both training and rendering. In contrast, another line of research addresses this by augmenting 3D Gaussians with a temporal axis to form 4D Gaussians [26,45,46,48], whose temporal slice at any timestamp reduces to a 3D distribution. We adopt this latter 4D Gaussian representation.

2.2 Geometry Estimation from Uncalibrated Casual Videos

3D Gaussian reconstruction pipelines heavily rely on Structure-from-Motion (SfM) [38] to obtain camera parameters and a sparse point cloud. However, this is

impractical for casually captured monocular videos that contain dynamic scenes, because SfM struggles to disambiguate camera motion from object motion. Fortunately, advances in visual foundation models have made it easier to jointly estimate scene geometry and camera parameters for dynamic scenes. DUSt3R [41] leverages its broadly learned priors to jointly estimate a point cloud of a static scene along with camera parameters. MonST3R [52] extends DUSt3R to dynamic scenes through additional dynamic supervision. More recently, VGGT-based approaches [4, 15, 40] achieve greater accuracy and robustness by scaling up both training data and model size. Nevertheless, the substantial GPU memory required for inference with these large networks limits their applicability to sequences containing hundreds of frames. In contrast to these fully data-driven methods, deep visual SLAM approaches like MegaSaM [23] offer a balanced solution. MegaSaM replaces handcrafted heuristics with differentiable bundle adjustment layers and learned priors, granting it superior robustness in dynamic scenes compared to traditional SLAM [29]. Crucially, its design that couples multi-stage feedforward inference with post-optimization ensures manageable computational costs even for long sequences. Considering the need for efficiency and robustness in processing extended monocular videos, we adopt MegaSaM for preprocessing our input videos.

2.3 Efficient Optimization for Gaussian Splatting

Although 3DGS offers an explicit representation and fast differentiable rasterization, its optimization still requires tens of minutes for a small scene. This has motivated recent research aimed at accelerating the optimization process. 3DGS-LM [12] replaces the Adam optimizer with Levenberg–Marquardt for faster convergence. [8, 10, 35] use exact tile–Gaussian intersection to reduce Gaussian–tile pairs and accelerate rasterization. Beyond that, reducing Gaussian redundancy has emerged as a key direction for improving training speed. Taming-3DGS [27] controls the number of densified Gaussians by computing importance scores from Gaussian attributes. DashGaussian [5] designs a resolution-guided scheduler to synchronize resolution with Gaussian count, thus suppressing Gaussian growth. SpeedySplat [10] removes Gaussians by computing Gaussian-related Hessian approximations. FastGS [37] introduces densification and pruning strategies based on multi-view consistency to optimize solely according to the importance of each Gaussian. However, these redundancy reduction strategies are specifically designed for static 3D Gaussians. In contrast, we propose a novel density control strategy tailored explicitly for 4D Gaussians.

3 The Proposed Minute4D

Given an unconstrained video $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^{N}$ where N is the number of frames, Minute4D reconstructs a 4D scene representation and render novel views in real-time from arbitrary views $\mathbf{P}^* \in \mathrm{SE}(3)$ at given timestamps t^* using Gaussian Splatting. Our pipeline is illustrated in Fig. 2. To begin with, we introduce the

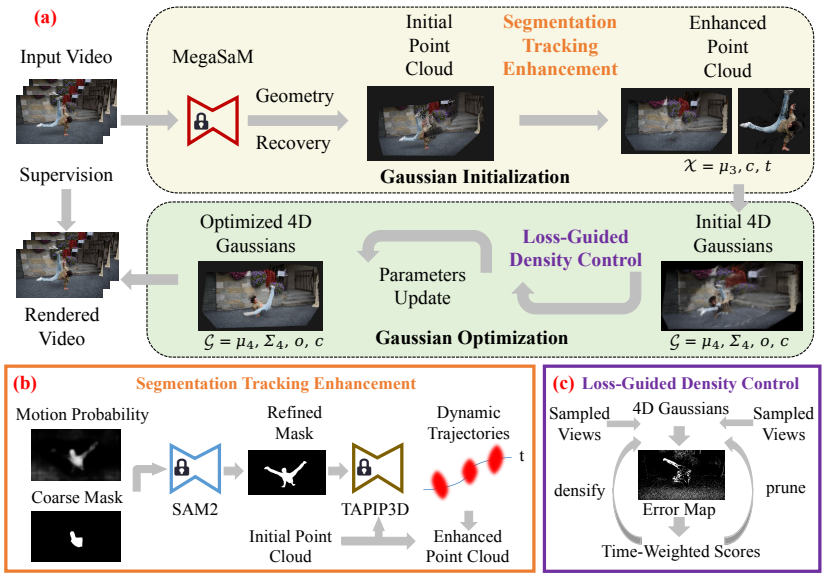

Fig. 2: Overview of our Minute4D framework: (a) We first estimate an initial point cloud for the dynamic scene from an unconstrained monocular video using MegaSaM [23]. This point cloud is then enhanced by our Segmentation-Tracking Enhancement module to obtain more complete geometry for Gaussian initialization. During Gaussian optimization, a novel Loss-Guided Density Control strategy is applied to reduce redundancy and accelerate the process. (b) The Segmentation-Tracking Enhancement module begins by binarizing MegaSaM's motion probability maps into coarse masks, which are subsequently refined with SAM2 [36]. These refined masks isolate dynamic points in the initial point cloud. Their trajectories are then estimated using TAPIP3D [51] and merged back, yielding the enhanced point cloud. (c) During the Loss-Guided Density Control stage, per-frame error maps are first computed based on the photometric $L1$ loss. From these error maps, time-weighted scores for Gaussians are derived, which subsequently guide the densification and pruning of Gaussians.

background of 4DGS for reconstruction from casual videos in Sec. 3.1. After that, we introduce our method and detail two key elements. Firstly, to enhance reconstruction fidelity, we propose a Segmentation-Tracking Enhancement module in Sec. 3.2. Secondly, to reduce redundancy and accelerate Gaussian optimization, we propose a Loss-Guided Density Control strategy in Sec. 3.3.

3.1 4DGS for Reconstruction from Casual Videos

4D Gaussians as Scene Representation. 4D Gaussian Splatting [26,48] represents a scene with a set of time-varying 3D Gaussians. Specifically, a standard 3D Gaussian is parameterized by its center $\mu \in \mathbb{R}^3$, covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$,

opacity $o \in \mathbb{R}^1$ as follows:

$$G(\mathbf{x}, \mu, \mathbf{\Sigma}, o) = o \exp\left(-\frac{1}{2}\left(\mathbf{x}-\mu\right)^\top \mathbf{\Sigma}^{-1}\left(\mathbf{x}-\mu\right)\right), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$ is a spatial point. 4DGS extends the standard 3D Gaussian to 4D by introducing an additional temporal dimension. This means that $\mathbf{x} \in \mathbb{R}^3$ is extended to $\mathbf{x}' \in \mathbb{R}^4$, $\mu \in \mathbb{R}^3$ is extended to $\mu' \in \mathbb{R}^4$, and $\mathbf{\Sigma} \in \mathbb{R}^{3\times3}$ is extended to $\mathbf{\Sigma}' \in \mathbb{R}^{4\times4}$. In addition, to represent a view- and time-dependent appearance, 4DGS employs a set of 4D spherindrical harmonics (4DSH), constructed by combining 3D spherical harmonics (SH) with temporal Fourier basis functions:

$$Z_{nl}^m\left(t, \theta, \phi\right) = \cos\left(\frac{2\pi n}{T} t\right) Y_l^m\left(\theta, \phi\right), \tag{2}$$

where Y_l^m is the 3D spherical harmonics. The index $l \geq 0$ denotes its degree, and m is the order satisfying $-l \leq m \leq l$. The index n denotes the temporal frequency and T denotes the temporal period.

Gaussian Initialization via Deep Visual SLAM. In casual videos, dynamic objects pose significant challenges to SfM, often making SfM-derived camera parameters unreliable. Inspired by [26], we instead employ a deep visual SLAM method, MegaSaM [23], to estimate camera parameters from casual videos, while jointly predicting depth maps. Specifically, as a video $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^N$ is input, MegaSaM outputs camera extrinsics $\mathcal{P} = \{\mathbf{P}_i \in \mathrm{SE}(3)\}_{i=1}^N$, intrinsics $\mathbf{K} \in \mathbb{R}^{3\times3}$, temporally consistent depth maps $\mathcal{D} = \{\mathbf{D}_i\}_{i=1}^N$, as well as motion probability maps $\hat{\mathcal{M}} = \{\hat{\mathbf{M}}_i\}_{i=1}^N$. Denoting MegaSaM as f_θ, we have:

$$\left(\mathcal{P}, \mathbf{K}, \mathcal{D}, \hat{\mathcal{M}}\right) = f_\theta(\mathcal{I}). \tag{3}$$

Among these estimates, motion probability maps $\hat{\mathcal{M}}$ are used to separate dynamic objects and static backgrounds, which will be detailed in Sec. 3.2. Based on the estimates above, each 2D pixel $\mathbf{p} \in \mathbb{R}^2$ is back-projected from image coordinates into 3D world coordinates $\mathbf{x} \in \mathbb{R}^3$:

$$\mathbf{x} = \mathbf{P}_i \circ \pi^{-1}\left(\mathbf{p}, \mathbf{D}_i, \mathbf{K}^{-1}\right), \quad \text{if} \quad \mathbf{p} \in \mathbf{I}_i, \tag{4}$$

where π^{-1} denotes the back-projection of pixel $\mathbf{p}$ into camera coordinates, and $\mathbf{P}_i$ transforms the 3D point into world coordinates. This operation yields a dense colored point cloud of the scene. Initializing Gaussians directly from this point cloud would produce excessive redundancy, so pruning is applied to the point cloud. First, the world space occupied by the point cloud is partitioned into a regular voxel grid, with the edge length adapted to the scene scale. After that, points within the same voxel cell are aggregated using a hash map and replaced by their centroid. Finally, the processed point cloud provides positions and colors for the initial Gaussians.

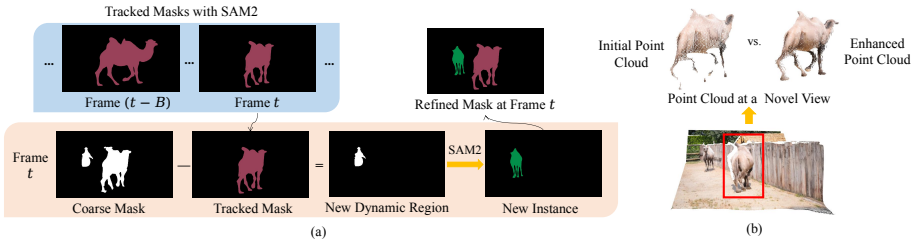

Fig. 3: Illustration of our Segmentation-Tracking Enhancement module. (a) The process of obtaining refined dynamic masks across video frames. To capture new instances appearing in later frames, new dynamic regions are identified by subtracting the tracked mask from the coarse mask, and are then refined by SAM2. (b) Comparison between the initial basic point cloud and the enhanced point cloud for a dynamic object (camel). The refined point cloud exhibits more complete geometry and finer detail.

3.2 Segmentation-Tracking Enhancement

Separating dynamic objects from the static background enables more accurate and robust initialization of Gaussians, which is key to achieving high reconstruction quality. Simply binarizing the motion probability maps $\hat{\mathcal{M}} = \{\hat{\mathbf{M}}_i\}_{i=1}^N$ by applying a fixed threshold τ_{motion} to obtain coarse dynamic masks $\mathcal{M}^c = \{\mathbf{M}_i^c\}_{i=1}^N$ often yields inaccurate results, as illustrated in Fig. 2 (b). Therefore, we refine the coarse dynamic masks using a 2D visual foundation model SAM2 [36]. Concretely, we first sample dynamic pixels from the coarse dynamic mask of the first frame $\mathbf{M}_1^c$ as prompts. These prompts, together with the full video $\mathcal{I}$, are input to SAM2 to perform instance segmentation, and the resulting masks are tracked across all frames. To capture dynamic objects that appear only in later frames, we also uniformly sample coarse masks from subsequent frames at an interval of B frames, and apply SAM2 again to segment newly emerging instances. The final refined dynamic masks $\mathcal{M}^r$ are obtained by integrating all segmented and tracked instances. The detailed procedure is illustrated in Fig. 3 (a).

After separating dynamic and static regions, we focus on augmenting dynamic points for better Gaussian initialization. Note that the point map of each frame is confined to the camera-visible regions and cannot populate occluded areas, while dynamic objects frequently self-occlude. Fortunately, different parts of a dynamic object become visible across frames, establishing correspondences for dynamic points is crucial to reconstruct regions that are missing in any single frame. To this end, we use a pre-trained 3D point-tracking model TAPIP3D [51], denoted by f_ϕ, to estimate trajectories of 3D dynamic points for each frame. Specifically, given the per-frame point maps $\mathcal{X} = \{\mathbf{X}_i\}_{i=1}^N$ and a 3D query point $\mathbf{y}$ (with its known position $\mathbf{y}_{i^*}$ at timestamp i^*), we use TAPIP3D to predict the position of this point across the entire sequence, yielding its full trajectory $\{\mathbf{y}_i\}_{i=1}^N$, which can be formulated as:

$$\{\mathbf{y}_i\}_{i=1}^N = f_\phi(\mathcal{X}, \mathbf{y}_{i^*}), \quad i^* \in 1, 2, \ldots, N. \tag{5}$$

For efficiency, we avoid dense querying of all dynamic points. Instead, we sample them at a regular interval and process multiple query points in batches by applying Eq. 5. The predicted trajectories $\underset{\mathbf{y}}{\cup}\{\mathbf{y}_i\}_{i=1}^{N}$ of these sampled points are then combined with the initial point cloud $\mathcal{X}$. By leveraging these predicted trajectories, we fill the occluded regions as much as possible in each frame, resulting in an enhanced point cloud $\mathcal{X}' = \left(\underset{\mathbf{y}}{\cup}\{\mathbf{y}_i\}_{i=1}^{N}\right) \cup \mathcal{X}$, as shown in Fig. 3 (b).

3.3 Loss-Guided Density Control

Following Gaussian initialization, Gaussians are optimized via photometric loss using the input video frames as supervision. A critical aspect of this optimization is adaptive population control of the Gaussians, through densification and pruning, which directly governs their number and therefore has a decisive impact on reconstruction efficiency.

In vanilla 4DGS [48], whether a Gaussian should be densified is determined solely by the magnitude of its gradient in a single frame. This causes Gaussian redundancy, which not only prevents faster training but also makes the reconstruction prone to artifacts. To avoid Gaussian redundancy, we aggregate a Gaussian's influence across multi-views when deciding whether to densify it, rather than relying on a single view. Moreover, to improve the efficiency of this decision process, we use the mean absolute error as the criterion instead of gradients. Concretely, at each densification step, K frames are firstly sampled from the video frame set $\mathcal{I}$, from which the ground-truth images $\bar{\mathcal{I}} = \{\bar{\mathbf{I}}_i\}_{i=1}^{K}$ and the corresponding rendered images $\hat{\mathcal{I}} = \{\hat{\mathbf{I}}_i\}_{i=1}^{K}$ are obtained. Then the per-pixel error is computed between each rendered image and its ground truth:

$$e_i^{u,v} = \mathbb{E}_c \left| \hat{\mathbf{I}}_i^{u,v} - \bar{\mathbf{I}}_i^{u,v} \right|, \tag{6}$$

where u, v denote pixel coordinates, and c denotes the color channel. After that, we apply a min–max normalization function to each frame's error map $\mathbf{E}_i$, which is then binarized with a predefined threshold τ_{error} to yield a mask:

$$\mathbf{E}_i^{mask} = \mathbb{I}\left(\mathit{Normalize}\left(\mathbf{E}_i\right) > \tau_{error}\right). \tag{7}$$

Here $\mathbb{I}$ denotes the indicator function and *Normalize* is the min-max normalization function. Pixels with value 1 in $\mathbf{E}_i^{mask}$ indicate regions of relatively high error. Employing $\mathbf{E}_i^{mask}$, we suppress relatively low-error pixels and retain only the high-error ones. The refined error map $\mathbf{E}_i^{refined}$ for each frame is then obtained via an element-wise multiplication of the original error map $\mathbf{E}_i$ with its corresponding mask:

$$\mathbf{E}_i^{refined} = \mathbf{E}_i \odot \mathbf{E}_i^{mask}, \tag{8}$$

where $\odot$ denotes Hadamard multiplication. The construction of refined error maps $\{\mathbf{E}_i^{refined}\}_{i=1}^{K}$ integrates multi-view information while preserving absolute inter-frame error differences, thereby enhancing the identification of Gaussians

requiring densification. Subsequently, We project each Gaussian $\mathcal{G}^j$ onto the refined error maps $\{\mathbf{E}_i^{refined}\}_{i=1}^K$ to obtain its 2D footprint $\{\mathbf{\Omega}_i^j\}_{i=1}^K$. And the sum of errors for all pixels covered by $\{\mathbf{\Omega}_i^j\}_{i=1}^K$ is defined as the densification score, denoted s_{den}^j. Considering that a 4D Gaussian's marginal distribution along the temporal dimension follows a one-dimensional normal distribution, the same 4D Gaussian should have different effects on different frames. To emphasize the differences in a 4D Gaussian's contributions across frames, we apply different weights to the error maps from different frames when computing s_{den}^j based on their temporal proximity. Frames close to the mean of the temporal normal distribution are assigned higher weights, while frames farther from the mean receive lower weights. It can be expressed as the following equation:

$$s_{den}^j = \frac{1}{K} \sum_{i=1}^{K} \sum_{\mathbf{q} \in \Omega_i^j} p^j(i) \cdot \mathbf{E}_i^{refined}(\mathbf{q}). \tag{9}$$

Here $p^j(\cdot)$ denotes the probability density function of Gaussian $\mathcal{G}^j$'s marginal distribution along the temporal dimension, μ^j is the mean of that marginal distribution, and $\mathbf{E}_i^{refined}(\mathbf{q})$ denotes the value of $\mathbf{E}_i^{refined}$ at pixel $\mathbf{q}$. A larger s_{den}^j indicates that Gaussian $\mathcal{G}^j$ affects a greater number of high-error pixels and consequently holds higher priority for densification. Given a threshold τ_{den}, Gaussians satisfying $s_{den}^j > \tau_{den}$ are marked for densification.

The vanilla 4DGS [48] prunes Gaussians with very low opacity or very large scale, but redundancy still persists. Therefore, we employ a similar procedure for selecting Gaussians to be pruned as that used for densification. Specifically, we randomly select K frames and compute the refined error maps $\mathbf{E}_i^{refined}$ refined in the same way. Since the photometric loss, used in Gaussian optimization, reliably indicates reconstruction quality, we incorporate it as a factor in the selection process. We define the photometric loss as $L^i = (1-\lambda) L_1^i + \lambda L_{SSIM}^i$, where L_1^i and L_{SSIM}^i denote the mean absolute error and the structural similarity loss, respectively, and λ is a hyperparameter. The pruning score s_{pru}^j of Gaussian $\mathcal{G}^j$ is computed analogously to the densification score s_{den}^j. The specific formula is as follows:

$$s_{pru}^j = Normalize\left(\sum_{i=1}^{K}\left(\sum_{\mathbf{q} \in \Omega_i^j} p^j(i) \cdot \mathbf{E}_i^{refined}(\mathbf{q})\right) \cdot L^i\right). \tag{10}$$

Here the photometric loss reflects the overall reconstruction degradation per frame. Accordingly, using it as a weighting factor effectively quantifies each Gaussian's contribution to this degradation. We similarly set a threshold τ_{pru}, abd any Gaussian with the pruning score $s_{pru}^j > \tau_{pru}$ is identified for pruning.

4 Experiments

In this section, we describe our experimental setup, evaluate our approach on novel view synthesis and optimization time, and then conduct ablation studies

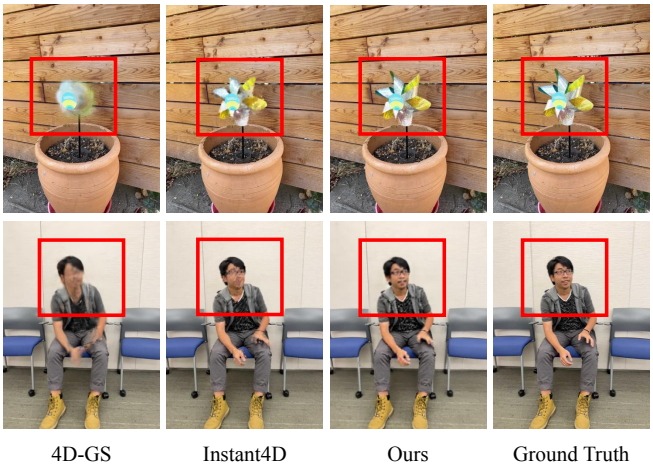

Fig. 4: Qualitative results on Dycheck iPhone dataset. Our method outperforms other competitive approaches in novel view synthesis, delivering superior rendering quality. It renders clearer details (e.g., the facial features in the second row) and more complete reconstructions (e.g., the paper windmill in the first row).

to validate our design choices. We clarify that all reported **runtimes** denote the time for Gaussian-based 4D scene optimization (or NeRF-based alternatives), with data preprocessing costs detailed in Tab. S1 of the supplementary material. Owing to page limitations, further evaluations are provided in the supplementary material.

4.1 Experimental Setup

Datasets. Our experiments are conducted on Dycheck iPhone dataset [9] and NVIDIA Dynamic dataset [50]. DyCheck iPhone dataset features generic, diverse dynamic scenes captured with a handheld iPhone using realistic camera motions. Its significant object motion and parallax pose a stringent challenge for dynamic reconstruction. NVIDIA Dynamic dataset includes seven dynamic sequences captured by twelve static cameras. Monocular input is simulated by selecting one frame from each consecutive camera, yielding twelve spatially and temporally sparse frames. Evaluation uses images from the first camera for testing. In addition, we evaluate our method on the in-the-wild DAVIS dataset [32], which features non-rigid motion, occlusions, and complex dynamics, thereby demonstrating its robustness to more challenging conditions.

Implementation Details. In the Segmentation-Tracking Enhancement module, the motion probability threshold τ_{motion} is set to 0.7. And the interval B is set to 50, 2 and 10 for Dycheck iPhone, NVIDIA Dynamic and DAVIS datasets, respectively. In 3D point tracking, we sample frames at intervals of 10, 1 and

PSNR↑	Apple	Block	Paper	Spin	Teddy	Average	Runtime(h)	Mem(GB)
D-NeRF [34]	24.23	21.80	21.85	22.15	19.46	21.50	> 24	12
RoDynRF [24]	17.38	15.99	20.71	16.66	13.28	16.80	22	15
4D-GS [42]	23.24	22.05	21.03	22.99	18.89	21.64	1.2	21
Deform3D [49]	24.82	23.26	20.62	23.51	20.93	22.63	–	–
RoDynRF(w/o pose) [24]	14.50	14.73	17.94	15.75	11.56	14.90	22	15
RoDyGS [16]	16.79	17.67	19.20	18.47	14.69	17.37	1.0	–
Instant4D(Lite) [26]	24.90	23.48	23.18	23.60	19.96	23.02	<u>0.03</u>	<u>1.1</u>
Instant4D(Full) [26]	<u>26.84</u>	<u>23.98</u>	<u>24.77</u>	<u>25.25</u>	<u>21.78</u>	<u>24.52</u>	0.12	8
Ours	**27.44**	**27.19**	**26.17**	**25.92**	**22.12**	**25.77**	**0.008**	**1.0**

Table 1: Quantitative results on Dycheck iPhone dataset. Methods above the midrule use ground-truth camera poses; those below do not require calibrated poses. *Runtime* indicates the **average training time** per scene, and *Mem* denotes the **peak GPU memory** consumption during optimization. Our method achieves the best PSNR scores while being significantly faster (0.008 h vs. 0.03-24 h for 4D Gaussian training) and more memory-efficient (1.0 GB vs. 1.1-21 GB) than all baselines.

10 for the three datasets. In the Loss-Guided Density Control module, we set $K = 10$ for frame sampling, with thresholds $\tau_{error} = 0.1$, $\tau_{den} = 0.001$ and $\tau_{pru} = 0.009$. During optimization, the maximum iteration numbers are set to 5,000 for Dycheck iPhone and DAVIS dataset, and 1,500 for NVIDIA Dynamic dataset, following the setup of [26]. All experiments are conducted on a single NVIDIA A6000 GPU.

4.2 Results

Results on Dycheck iPhone dataset. We evaluate Minute4D against several competitive methods on Dycheck iPhone dataset following the evaluation protocol of [16]. Quantitatively, we evaluate our method across three key metrics: rendering quality (measured by PSNR, where higher is better), runtime (lower is better), and peak GPU memory usage (lower is better). As shown in Tab. 1, Minute4D achieves the highest PSNR on every scene of the Dycheck iPhone dataset, outperforming all other methods—including those that use ground-truth camera poses. Critically, this significant improvement in rendering quality (an average gain of **1.25 dB**) is accomplished concurrently with a dramatic reduction in computational resource requirements. Our method completes training **1.32 minutes faster** and consumes **10**% less peak GPU memory than the leading baseline. These results demonstrate a simultaneous advance in both reconstruction fidelity and efficiency. We also qualitatively compares the quality of novel view synthesis on Dycheck iPhone dataset. As shown in Fig. 4, Minute4D's reconstruction demonstrates significantly higher fidelity, as evidenced by its excellence in both structural integrity and fine detail preservation. For instance, in the first row, the paper windmill reconstructed by our approach maintains sharper edges and more coherent geometry compared to the uncomplete or blurred outputs from 4D-GS [42] and Instant4D [26]. And in the second row, the facial features rendered by our method exhibit significantly higher fidelity, with less artifacting

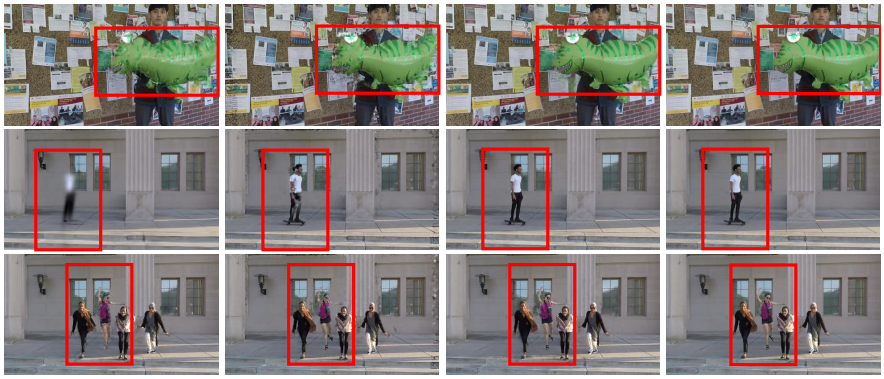

Fig. 5: Qualitative results on NVIDIA Dynamic dataset. Whether facing a violently shaking dynamic object (first row), lateral human motion (second row), or vertical jumping (third row), our method captures and reconstructs the motion more accurately, achieving superior novel view synthesis.

than the other two methods. The qualitative results affirm that our approach better preserves both high-frequency details and overall scene consistency in novel view synthesis.

Results on NVIDIA Dynamic dataset. We evaluate Minute4D on NVIDIA Dynamic dataset following the protocol of [24]. In addition to rendering quality and runtime, we also compare rendering speed (measured in frame rate, higher is better). The quantitative results in Tab. 2 highlight the superior performance of our method. Among all approaches that rely on visual SLAM for camera estimation (pose-free setting), our method achieves the highest rendering quality (PSNR of **24.73 dB**), surpassing the previous best pose-free method by **0.74 dB**. Notably, this superior quality is achieved with a drastically reduced computational footprint: training concludes in merely **0.006 hours** while also attaining the top rendering speed of **1218 FPS**. Compared to methods that require offline calibration (e.g., COLMAP), our approach delivers competitive rendering fidelity while being orders of magnitude faster in both reconstruction and rendering, demonstrating an effective balance between high fidelity and practical efficiency for dynamic novel view synthesis. In addition, Fig. 5 provides a qualitative comparison on NVIDIA Dynamic dataset. Minute4D preserves the better structure and temporal coherence of fast-moving objects (the first row), maintains clearer subject boundaries during translation (the second row), and effectively reconstructs the motion and shape of dynamic human figures(the third row), with fewer artifacts or blurring compared to other approaches.

Qualitative results on DAVIS dataset. To validate Minute4D's robustness under more challenging conditions, we perform qualitative evaluations on the in-

Method	Calibration	Runtime(h) ↓	Rendering FPS ↑	PSNR ↑
HyperNeRF [31]	COLMAP	64	0.40	17.60
DynamicNeRF [9]	COLMAP	74	0.05	**26.10**
RoDynRF [24]	COLMAP	28	0.42	<u>25.89</u>
4D-GS [42]	COLMAP	1.2	43	21.45
Casual-FVS [20]	Video-Depth-Pose	0.25	48	24.57
InstantSplat* [7]	Visual SLAM	0.15	117	22.56
4DGS* [48]	Visual SLAM	0.16	98	18.34
Instant4D [26]	Visual SLAM	<u>0.02</u>	<u>822</u>	23.99
Ours	Visual SLAM	**0.006**	**1218**	24.73

Table 2: Quantitative comparison on NVIDIA Dynamic Scene dataset. "Runtime" follows the definition in Tab. 1. Methods denoted with an asterisk * use Visual SLAM for camera calibration instead of their original COLMAP-based pipeline, ensuring a fair comparison in the pose-free setting. Our method achieves the best rendering quality among all pose-free methods while simultaneously reaching the fastest training speed (0.006 h) and rendering rate (1218 FPS), establishing a new state-of-the-art in efficient, high-fidelity dynamic scene reconstruction.

Method	PSNR ↑	SSIM ↑	Runtime(sec) ↓	Mem(MB) ↓
w/o Seg-Track	24.60	0.628	<u>31.78</u>	**1064**
w/o Loss-Guided	<u>25.74</u>	<u>0.694</u>	397.732	9406
Full Settings	**25.77**	**0.717**	**29.224**	<u>1088</u>

Table 3: Ablations on Dycheck iPhone dataset. Here, "Runtime" and "Mem" follow the definitions in Tab. 1. "Seg-Track" denotes the Segmentation-Tracking Enhancement module, and "Loss-Guided" denotes the Loss-Guided Density Control strategy. Removing the Segmentation-Tracking Enhancement module degrades quality (PSNR/SSIM). Disabling Loss-Guided Density Control drastically increases runtime and memory. Our full model achieves the best performance in terms of both speed and quality.

the-wild DAVIS dataset, known for its complex dynamics. For each scene in this dataset, the entire video sequence is used as input. During novel view synthesis, the camera view at each timestamp is fixed to that of the first frame, following the same protocol as used for NVIDIA Dynamic dataset. Fig. 6 presents qualitative results of our method, which show that Minute4D achieves the best reconstruction quality across the diverse and challenging scenes in DAVIS dataset. Moreover, our method trains rapidly on DAVIS: for instance, the Breakdance-flare sequence, comprising 72 images, is reconstructed in only **42.50s** —significantly faster than the **145.42s** required by Instant4D.

4.3 Ablation Study

To further demonstrate the effectiveness of our method, we conduct ablation studies with or without certain components, *i.e.*, Segmentation-Tracking En-

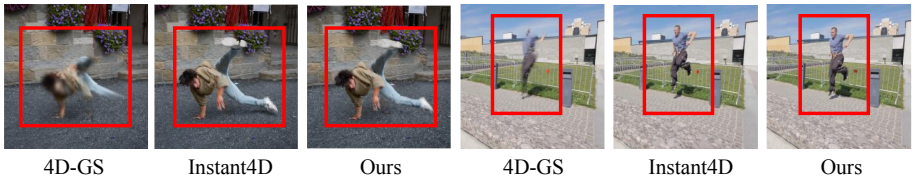

Fig. 6: Visualization on DAVIS dataset. For novel view synthesis, the reconstructed 4D Gaussians are rendered from the view of the first frame. Results show that Minute4D maintains superior reconstruction quality across the diverse and challenging scenes in DAVIS dataset.

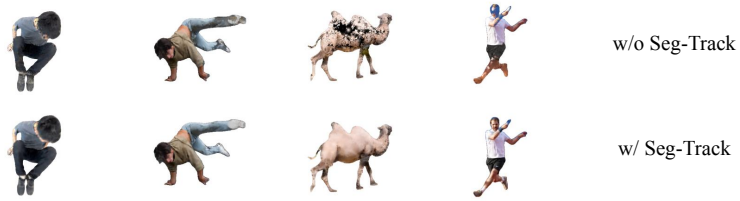

Fig. 7: Ablations on Dycheck iPhone dataset and DAVIS dataset. Here, "Seg-Track" denotes the Segmentation-Tracking Enhancement module. Removing this module causes holes and artifacts, resulting in a significant drop in rendering quality for dynamic objects.

hancement and Loss-Guided Density Control. Quantitative results on Dycheck iPhone dataset are provided in Tab. 3. More ablation studies are provided in the supplementary materials.

As demonstrated in Tab. 3, removing the Segmentation-Tracking Enhancement module results in a noticeable drop in reconstruction fidelity, confirming its critical role in recovering accurate geometry. Disabling the Loss-Guided Density Control strategy causes a drastic increase in runtime by over 13× and memory consumption by nearly 9×, although PSNR/SSIM remain largely unchanged. This highlights the strategy's essential function in pruning redundant Gaussians and maintaining efficiency. Our method with full settings achieves the best overall performance, showing that the two components operate complementarily: the Segmentation-Tracking module ensures high-quality Gaussian initialization, while the Loss-Guided Density Control strategy reduces Gaussian redundancy and enables efficient optimization. Together, they collectively achieve superior performance across fidelity, speed, and memory usage.

To more directly assess the impact of the Segmentation-Tracking module on the quality of reconstruction, we visualize the novel view synthesis of dynamic objects. As illustrated in Fig. 7, removing this module results in severe holes and artifacts, which significantly degrade the rendering fidelity.

5 Conclusion

In this paper, we present Minute4D, a 4D Gaussian Splatting method that reconstructs high-fidelity dynamic scenes from monocular video in one minute. Its quality and efficiency stem from two key components: a Segmentation-Tracking Enhancement module, which robustly isolates and completes dynamic object geometry for better Gaussian initialization, and a Loss-Guided Density Control strategy that reduces Gaussian redundancy guided by multi-view error information for higher optimization efficiency. Together, these two innovations enable state-of-the-art performance across several benchmarks in both reconstruction quality and optimization speed.

Acknowledgements

This work was supported by Beijing Natural Science Foundation under Grant No. L241012 and National Natural Science Foundation of China under Grant No. 62403459, No. 62572468.

References

1. Basha, T., Moses, Y., Kiryati, N.: Multi-view scene flow estimation: A view centered variational approach. International journal of computer vision **101**(1), 6–21 (2013)
2. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662). vol. 2, pp. 690–696. IEEE (2000)
3. Cao, A., Johnson, J.: Hexplane: A fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 130–141 (2023)
4. Chen, X., Xiong, Z., Chen, Y., Li, G., Wang, N., Luo, H., Chen, L., Sun, H., Wang, B., Chen, G., et al.: Dggt: Feedforward 4d reconstruction of dynamic driving scenes using unposed images. arXiv preprint arXiv:2512.03004 (2025)
5. Chen, Y., Jiang, J., Jiang, K., Tang, X., Li, Z., Liu, X., Nie, Y.: Dashgaussian: Optimizing 3d gaussian splatting in 200 seconds. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11146–11155 (2025)
6. Doersch, C., Luc, P., Yang, Y., Gokay, D., Koppula, S., Gupta, A., Heyward, J., Rocco, I., Goroshin, R., Carreira, J., et al.: Bootstap: Bootstrapped training for tracking-any-point. In: Proceedings of the Asian Conference on Computer Vision. pp. 3257–3274 (2024)
7. Fan, Z., Cong, W., Wen, K., Wang, K., Zhang, J., Ding, X., Xu, D., Ivanovic, B., Pavone, M., Pavlakos, G., et al.: Instantsplat: Sparse-view gaussian splatting in seconds. arXiv preprint arXiv:2403.20309 (2024)
8. Feng, G., Chen, S., Fu, R., Liao, Z., Wang, Y., Liu, T., Hu, B., Xu, L., Pei, Z., Li, H., et al.: Flashgs: Efficient 3d gaussian splatting for large-scale and high-resolution rendering. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26652–26662 (2025)
9. Gao, H., Li, R., Tulsiani, S., Russell, B., Kanazawa, A.: Monocular dynamic view synthesis: A reality check. Advances in Neural Information Processing Systems **35**, 33768–33780 (2022)
10. Hanson, A., Tu, A., Lin, G., Singla, V., Zwicker, M., Goldstein, T.: Speedy-splat: Fast 3d gaussian splatting with sparse pixels and sparse primitives. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21537–21546 (2025)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
12. Höllein, L., Božič, A., Zollhöfer, M., Nießner, M.: 3dgs-lm: Faster gaussian-splatting optimization with levenberg-marquardt. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 26740–26750 (2025)
13. Hu, M., Yin, W., Zhang, C., Cai, Z., Long, X., Chen, H., Wang, K., Yu, G., Shen, C., Shen, S.: Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(12), 10579–10596 (2024)
14. Hu, W., Gao, X., Li, X., Zhao, S., Cun, X., Zhang, Y., Quan, L., Shan, Y.: Depthcrafter: Generating consistent long depth sequences for open-world videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2005–2015 (2025)
15. Hu, Y., Cheng, C., Yu, S., Guo, X., Wang, H.: Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. arXiv preprint arXiv:2511.19971 (2025)

16. Jeong, Y., Lee, J., Choi, H., Cho, M.: Rodygs: Robust dynamic gaussian splatting for casual videos. arXiv preprint arXiv:2412.03077 (2024)
17. Karaev, N., Makarov, Y., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6013–6022 (2025)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
19. Kim, M., Lim, J., Han, B.: 4d gaussian splatting in the wild with uncertainty-aware regularization. Advances in Neural Information Processing Systems **37**, 129209–129226 (2024)
20. Lee, Y.C., Zhang, Z., Blackburn-Matzen, K., Niklaus, S., Zhang, J., Huang, J.B., Liu, F.: Fast view synthesis of casual videos with soup-of-planes. In: European Conference on Computer Vision. pp. 278–296. Springer (2024)
21. Lei, J., Weng, Y., Harley, A.W., Guibas, L., Daniilidis, K.: Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6165–6177 (2025)
22. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European conference on computer vision. pp. 71–91. Springer (2024)
23. Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., Snavely, N.: Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10486–10496 (2025)
24. Liu, Y.L., Gao, C., Meuleman, A., Tseng, H.Y., Saraf, A., Kim, C., Chuang, Y.Y., Kopf, J., Huang, J.B.: Robust dynamic radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13–23 (2023)
25. Liu, Y.L., Gao, C., Meuleman, A., Tseng, H.Y., Saraf, A., Kim, C., Chuang, Y.Y., Kopf, J., Huang, J.B.: Robust dynamic radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13–23 (2023)
26. Luo, Z., Ran, H., Lu, L.: Instant4d: 4d gaussian splatting in minutes. arXiv preprint arXiv:2510.01119 (2025)
27. Mallick, S.S., Goel, R., Kerbl, B., Steinberger, M., Carrasco, F.V., De La Torre, F.: Taming 3dgs: High-quality radiance fields with limited resources. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
28. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
29. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: A versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015)
30. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 343–352 (2015)
31. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. arXiv preprint arXiv:2106.13228 (2021)
32. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 724–732 (2016)

33. Piccinelli, L., Yang, Y.H., Sakaridis, C., Segu, M., Li, S., Van Gool, L., Yu, F.: Unidepth: Universal monocular metric depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10106–10116 (2024)
34. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10318–10327 (2021)
35. Radl, L., Steiner, M., Parger, M., Weinrauch, A., Kerbl, B., Steinberger, M.: Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering. ACM Transactions on Graphics (TOG) **43**(4), 1–17 (2024)
36. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024), `https://arxiv.org/abs/2408.00714`
37. Ren, S., Wen, T., Fang, Y., Lu, B.: Fastgs: Training 3d gaussian splatting in 100 seconds. arXiv preprint arXiv:2511.04283 (2025)
38. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
39. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
40. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
41. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20697–20709 (2024)
42. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20310–20320 (2024)
43. Xiao, Y., Wang, Q., Zhang, S., Xue, N., Peng, S., Shen, Y., Zhou, X.: Spatialtracker: Tracking any 2d pixels in 3d space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20406–20417 (2024)
44. Xu, J., Fan, Z., Yang, J., Xie, J.: Grid4d: 4d decomposed hash encoding for high-fidelity dynamic gaussian splatting. Advances in Neural Information Processing Systems **37**, 123787–123811 (2024)
45. Xu, Z., Peng, S., Lin, H., He, G., Sun, J., Shen, Y., Bao, H., Zhou, X.: 4k4d: Real-time 4d view synthesis at 4k resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20029–20040 (2024)
46. Xu, Z., Xu, Y., Yu, Z., Peng, S., Sun, J., Bao, H., Zhou, X.: Representing long volumetric video with temporal gaussian hierarchy. ACM Transactions on Graphics (TOG) **43**(6), 1–18 (2024)
47. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos. arXiv preprint arXiv:2304.11968 (2023)
48. Yang, Z., Yang, H., Pan, Z., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint arXiv:2310.10642 (2023)

49. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20331–20341 (2024)
50. Yoon, J.S., Kim, K., Gallo, O., Park, H.S., Kautz, J.: Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5336–5345 (2020)
51. Zhang, B., Ke, L., Harley, A.W., Fragkiadaki, K.: Tapip3d: Tracking any point in persistent 3d geometry. arXiv preprint arXiv:2504.14717 (2025)
52. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. arXiv preprint arXiv:2410.03825 (2024)