Racing in Volume with Flow Ensembles

Saswat Subhajyoti Mallick, Riu Cherdchusakulchai, Marc Ruiz Olle, Albert Mosella-Montoro, Jose Ribeiro-Gomes, Francisco Vicente Carrasco, Fernando De la Torre

Carnegie Mellon University, USA
humansensinglab.github.io/monaco4d

Fig. 1: Given multi-view video streams, **FastFlowGS** reconstructs geometry for high fidelity, photorealistic novel-view rendering. We introduce **Monaco4D**, the first synthetic outdoor dataset, with Formula 1 racing scenarios to benchmark extreme motion.

Abstract. Streaming 4D reconstruction has been demonstrated only indoors, on dense camera rigs surrounding subjects that move at human pace. Outdoor 4D reconstruction exists but relies either on cameras mounted on the moving vehicle itself, or on limited-coverage arrays observing quasi-static subjects offline. The case that actually matters for spectators is a fast-moving subject, watched from a sparse ring of allocentric cameras, streaming. No method targets this, and no benchmark exists to evaluate one. To this end, we introduce **FastFlowGS**, a streaming 4D Gaussian Splatting method for reconstructing fast-moving subjects from a small set of fixed external cameras, and **Monaco4D**, a photorealistic Unreal Engine 5 benchmark for high-speed outdoor reconstruction. FastFlowGS fuses sparse matches, semi-dense tracks, and dense optical flow by lifting each signal to 3D with geometric uncertainty and combining them through a Kalman-style temporal update. **Monaco4D** provides

Formula 1 sequences under varied illumination from trackside, onboard, and drone viewpoints with dense ground truth. On CMU-Panoptic (a public dataset), FastFlowGS exceeds the strongest baseline by 12.6% VMAF at 35% greater efficiency. On Monaco4D, where existing streaming methods degrade severely, it improves dynamic-region PSNR by up to 18.6% with 28.3% lower per-frame optimization time.

Keywords: 4D Reconstruction · Novel view synthesis · Immersive Sports

1 Introduction

"Auto racing began five minutes after the second car was built." ∼ Henry Ford

The instinct to race is older than the sport itself, and no series pushes it further than Formula 1. Over 820 million fans follow F1, yet nearly all of them watch through a single director-chosen frame that compresses a spatial spectacle into a confined, flat image. VR and AR offer a way out, placing the viewer anywhere in the scene, but they demand real-time 4D reconstruction of dynamic outdoor environments, a challenge that no sport pushes harder than Formula 1.

Cars pass trackside cameras at over 300 km/h, their carbon-fibre bodywork specularly reflecting the environment, while lighting shifts abruptly inside tunnels. The core difficulty is not speed itself but the pixel displacements it produces. Filmed laterally at 30 Hz, a Formula 1 car moves 200 to 400 pixels between consecutive frames, an order of magnitude beyond what optical flow [25, 42] and point trackers [6, 20] handle reliably.

Existing methods take one of two approaches, and neither holds at this scale. Deformable representations [4, 37, 48, 53] assume slow or camera-dominated motion and degrade when displacements grow large [43]. Streaming methods [11, 40, 50, 54] each commit to a single pixel-space correspondence mechanism, so when that mechanism fails, the error propagates directly into the reconstruction. Available benchmarks further reinforce the problem. Driving datasets [3, 31, 41] are captured from ego-vehicles and lack the lateral displacements of fixed trackside cameras, while indoor sequences [19, 28] have slow dynamics and controlled lighting. Together, they leave the combination of large pixel displacements, diverse exocentric viewpoints, and uncontrolled outdoor illumination largely unexplored.

Monaco4D is the first multi-view benchmark built around this gap. Rendered in Unreal Engine 5 [7] on a to-scale replica of the Monaco circuit with vehicle dynamics approximating real lap profiles, it provides seven sequences, five illumination conditions, three camera modalities (trackside, onboard, drone), and dense ground-truth annotations. We pair it with **FastFlowGS**, a streaming Gaussian Splatting method built on a simple insight that *no single correspondence mechanism is reliable under large displacements, but their disagreement reveals where each one fails.* FastFlowGS fuses sparse feature matches, semi-dense point tracks, and dense optical flow, using cross-level agreement scoring to identify which signal is trustworthy at each location. The reliable correspondences are lifted into 3D through uncertainty-aware triangulation and merged with a Kalman temporal prior, producing per-Gaussian positions that converge

at a fraction of the cost of standard streaming approaches.

Our contributions are:

1. **FastFlowGS**, a streaming 4D Gaussian reconstruction method that fuses correspondence signals across scales to handle large pixel displacements.
2. **Monaco4D**, the first multi-view outdoor dataset, featuring F1 sequences with extreme displacements, varied illumination, and camera modalities.
3. SOTA on CMU-Panoptic [19] (12.6% higher VMAF, 35% faster) and the first streaming reconstruction on Monaco4D, where prior methods fail entirely.

2 Related Work

Dynamic scene reconstruction has advanced rapidly in controlled indoor and egocentric driving settings, yet fast-moving outdoor subjects observed from spectator viewpoints remain largely unexplored.

Deformable Representations and Gaussian Playback. A common paradigm models dynamic scenes via a canonical representation with per-frame deformations. D-NeRF [37] pioneered deformable radiance fields, K-Planes [9] and HexPlane [4] introduced efficient explicit factorizations, Fourier PlenOctrees [45] and VideoRF [44] enable real-time rendering. All assume slow deformation or dominant camera motion and degrade under rapid object displacement [43]. Gaussian extensions [29, 30, 36, 48, 53, 58] achieve large speedups for offline playback but target studio or indoor sequences where per-frame displacements rarely exceed tens of pixels.

Streaming and Feed-Forward 4D Reconstruction. Streaming methods accelerate per-frame convergence through diverse strategies: Dynamic-3DGS [34] enforces persistent attributes with rigidity priors. 3DGStream [40] uses a Neural Transformation Cache; HiCoM [11] and ReCon-GS [10] use hierarchical grids with dynamic reconfiguration; QUEEN [12], GIFStream [27], and Instant Gaussian Stream [50] employ residual quantization, feature streams, and flow-based anchor control, respectively. TrackerSplat [54], the closest predecessor to FastFlowGS, initializes Gaussians from point trackers with per-frame finetuning; we extend it with multi-resolution fusion, uncertainty-aware triangulation, and a variational position prior. Feed-forward approaches [5, 15, 21, 24, 26, 46, 47, 49, 55–57] bypass test-time optimization via pointmap regression or persistent-state models but remain too compute-heavy for real-time use.

Outdoor Dynamic Scenes and Sports Reconstruction. EmerNeRF [52] decomposes Waymo sequences into static, dynamic, and flow fields; StreetSurf [14] and Street Gaussians [51] adapt implicit surfaces and 3DGS to forward-facing trajectories; SEED4D [22] adds exocentric views but remains urban and pedestrian-scale. All assume ego-motion with strong forward parallax, which breaks in spectator settings where objects move faster than the camera baseline. In sports, industrial systems [2, 13, 18, 39] reconstruct trajectories of compact objects (balls, shuttlecocks), not full 4D scenes. Free-viewpoint sports video dates to Kanade et

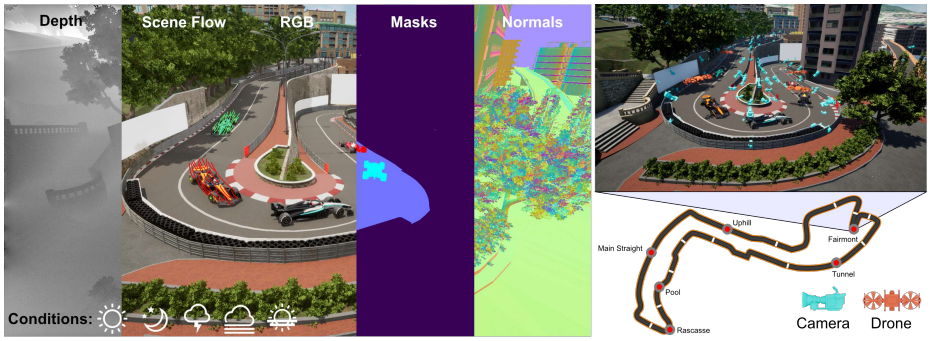

Fig. 2: Overview of the Monaco4D benchmark.: We cover six sections of the Monaco circuit with dense multi-view camera coverage, multimodal ground-truth annotations (RGB, depth, normals, masks, scene flow), under five illumination conditions.

al.'s Virtualized Reality dome [38]; AerialRecon [16] reconstructs outdoor athletes from a single drone, and LiveSplats [17] demonstrates real-time Gaussian reconstruction of indoor arenas. However, these systems operate under controlled lighting with subjects moving at up to 10 m/s; neither the displacement regime nor the illumination conditions of outdoor motorsport have been addressed.

3 Monaco4D Benchmark

Monaco4D is a synthetic multi-view dataset rendered in Unreal Engine 5 with path-traced photorealism. It comprises six sequences on a to-scale replica of the Monaco Grand Prix circuit, each capturing a distinct segment (shown in Fig. 4 in the supplemental) with single-car and multi-car variants that include realistic maneuvers such as overtaking, near-crashes, and formation drafting (Fig. 2).

Camera rig. Each sequence provides fixed *trackside cameras* that are positioned near the guardrails along the circuit, *onboard cameras* that follow FIA specifications [1] with seven per vehicle plus one helmet-mounted unit, covering forward, rear, lateral, and driver perspectives, and *drone cameras* follow three trajectories that simulate broadcast operation, including subject tracking, rapid panning, and abrupt zoom changes. More details are in Sec. 2.1 of the supplemental.

Illumination conditions. Each sequence is rendered under five conditions (day, evening, night, fog, rain), producing 30 sequence-illumination combinations before car-count variations. We also add within-clip transitions like tunnel entries, intermittent shade from buildings, and streetlamp pools at night, to challenge the slowly-varying illumination assumption of most photometric methods.

Ground-truth annotations. Each frame provides path-traced RGB, surface normals, metric depth, per-instance segmentation masks, and dense 3D scene flow.

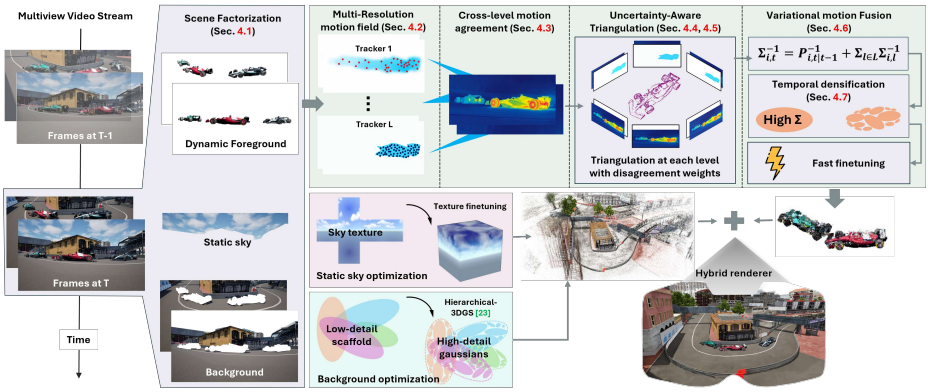

Fig. 3: Overview of FastFlowGS. We decompose the scene into sky, background, and foreground (Sec. 4.1). Sky and background are reconstructed via a skybox and Hierarchical 3DGS [23], respectively. Multi-resolution 2D motion fields are estimated per view (Sec. 4.2) and scored for cross-level agreement (Sec. 4.3). Gaussians are associated with their contributing pixels (Sec. 4.4) and each level's estimates are lifted to 3D via triangulation (Sec. 4.5). An uncertainty-based fusion merges all levels with a Kalman-based temporal prior into a single position estimate per Gaussian (Sec. 4.6), which seeds fast per-frame optimization and densification (Sec. 4.6). The three layers are composited at render time. Refer Alg. 1 in the supplemental for pseudocode.

4 FastFlowGS

We find that no single correspondence type is reliable across all surfaces and motions. Sparse features fail on textureless regions, dense flow degrades near occlusions, and any level may appear equally plausible where little motion occurs. FastFlowGS resolves this by measuring cross-level agreement as a proxy for reliability, lifting each level to 3D with a covariance estimate, and fusing all sources through a single information-form update where each contributes in proportion to its geometric precision.

4.1 Scene Factorization for Dynamic Reconstruction

We represent the scene as three independently fitted layers composited at render time: static background, dynamic foreground, and sky. The background uses Hierarchical 3DGS [23], optimized once at $t=0$. For subsequent frames, geometry is frozen and only SHs (color) are updated to account for illumination drift [17]. Dynamic Gaussians are initialized at $t=0$ by running 3DGS on a hemispherical rig around the cars with scale regularization [8, 35] For $t>0$, they are identified by projecting onto the previous frame's segmentation masks and repositioned via the fusion of Sec. 4.6. The sky violates finite-depth assumptions and is rendered as a low-resolution skybox [51] composited behind other layers.

4.2 Multi-Resolution Motion Field Estimation

FastFlowGS estimates per-view 2D motion fields by integrating correspondence sources at different spatial scales. An arbitrary number of trackers with varying levels of sparsity comprise the set L; since triangulation (Sec. 4.5) runs independently per level, all levels execute in parallel. Sparse tracks are keypoint correspondences [33], medium are semi-dense point tracks [6, 20], and dense are per-pixel optical flow [25, 42].

Voronoi densification. Sparse and medium tracks are densified to full resolution via Gaussian-weighted Voronoi interpolation [25]. For each pixel $\mathbf{p} = (u, v)$, the $K = 8$ nearest track observations are interpolated with weights $w_i \propto \exp(-d_i^2/2\sigma^2)$ ($\sigma = 0.05$), where d_i is the distance to Voronoi center i. The nearest-neighbor kernel value defines confidence $\mathbf{C}^{\text{vor}}(\mathbf{p})$, which down-weights pixels far from any track during triangulation. More details are in Sec. 1.4 and Alg. 2 of the supplemental.

4.3 Cross-Level Motion Agreement

Since each correspondence tracker has its limitations, rather than selecting a preferred level, we estimate reliability by measuring agreement between them: pixels where motion estimates disagree are downweighted regardless of which estimate is correct.

Formally, let $\mathbf{F} = \{\mathbf{F}_i\}$ where $i \in L$, $\mathbf{F}_i \in \mathbb{R}^{V\times H\times W\times 2}$ denote the densified flow fields for all tracking levels over V views. Each level's per-pixel confidence is:

$$\mathbf{C}_i = \psi(\mathbf{C}_i^{\text{vor}}, \mathbf{F}) = \mathbf{C}_i^{\text{vor}} \odot \exp\left(-\frac{1}{2} \cdot \frac{\sum_{k\neq i,\, k\in L} D_{ik}^2}{(\sigma\bar{\mathbf{F}})^2}\right) \tag{1}$$

$$\mathbf{C}_s = \psi(\mathbf{C}_s^{\text{vor}}, \mathbf{F}), \quad \mathbf{C}_m = \psi(\mathbf{C}_m^{\text{vor}}, \mathbf{F}), \quad \mathbf{C}_d = \psi(\mathbf{C}_d^{\text{vor}}, \mathbf{F}), \tag{2}$$

where $D_{ab} = \|\mathbf{f}_a - \mathbf{f}_b\|_2$ for $\{a, b\} \in L$, and $\bar{\mathbf{F}} = \frac{1}{|L|}\|\sum_{i\in L} \mathbf{F}_i\|_2$ is the mean flow magnitude, clamped below at 1 px to avoid amplifying static regions. Eqs. (1)–(2) encode two complementary principles: a track far from any observation is unreliable (Voronoi term), and a track that disagrees with the other levels is unreliable regardless of its density (cross-level term).

4.4 Gaussian-to-Pixel Association

To associate each Gaussian with its contributing pixels, we record the top-K Gaussian indices and alpha-blending weights $\alpha_{v,\mathbf{p},k}$ for every pixel $\mathbf{p}$ and view $v \in V$, yielding weighted observations $\{i, v, \mathbf{p}, \alpha_{v,p,i}\}$ used to locate each Gaussian in image space and weight its triangulation contribution (Sec. 4.5).

Each observation is weighted by:

$$w_{v,\mathbf{p},i,l} = \alpha_{v,\mathbf{p},i} \cdot \mathbf{C}_l(v, \mathbf{p}) \cdot \|\mathbf{F}_l(v, \mathbf{p})\|_2. \tag{3}$$

The product structure is such that, a Gaussian contributes to a position estimate only when it renders prominently (α), the correspondence is geometrically consistent ($\mathbf{C}_l$), and there is actual motion to track ($\|\mathbf{F}_l\|$). The projected mean of each Gaussian i is then updated per view v and tracker l as $\hat{\mu}^{2d}_{v,i,l} = \mu^{2d}_{v,i} + \sum_{\mathbf{p}} \alpha_{v,\mathbf{p},i}\mathbf{F}_l(v,\mathbf{p})$, to be used in triangulation.

4.5 Uncertainty-Aware Multi-View Triangulation

We estimate updated Gaussian positions $\hat{\mu}^{3d}_{i,l}$ as the weighted intersection of viewing rays, discarding views whose geometry is ambiguous.

Weighted DLT triangulation. Given projected observations $\{\hat{\mu}^{2d}_{v,i,l}\}$, we estimate each 3D position by minimizing the weighted reprojection error. For unit ray direction $\hat{\mathbf{d}}_{v,i,l}$ lifted from $\hat{\mu}^{2d}_{v,i,l}$ via camera intrinsics and extrinsics, the DLT constraint is:

$$\overbrace{\left(\sum_{\mathbf{p}} w_{v,\mathbf{p},i,l}\right)}^{\eta_{v,i,l}} \cdot (\mathbf{I}_{3\times 3} - \hat{\mathbf{d}}_{v,i,l}\hat{\mathbf{d}}^{\top}_{v,i,l})(\hat{\mu}^{3d}_{i,l} - \mathbf{o}_v) = \mathbf{0}_{3\times 1}, \tag{4}$$

where $\mathbf{o}_v$ is the camera center. For Gaussian i visible from N_i views, these constraints form a weighted least-squares system $\|\mathbf{W}[\mathbf{A}\hat{\mu}^{3d}_{i,l} - \mathbf{b}]\|_2^2 = \mathbf{0}$, where $\mathbf{W} = \mathrm{diag}(\eta_1\mathbf{I}_{3\times 3}, \ldots, \eta_{N_i}\mathbf{I}_{3\times 3})$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} - \hat{\mathbf{d}}_1\hat{\mathbf{d}}_1^{\top} \\ \vdots \\ \mathbf{I} - \hat{\mathbf{d}}_{N_i}\hat{\mathbf{d}}_{N_i}^{\top} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} (\mathbf{I} - \hat{\mathbf{d}}_1\hat{\mathbf{d}}_1^{\top})\mathbf{o}_1 \\ \vdots \\ (\mathbf{I} - \hat{\mathbf{d}}_{N_i}\hat{\mathbf{d}}_{N_i}^{\top})\mathbf{o}_{N_i} \end{bmatrix},$$

with closed-form solution $\hat{\mu}^{3d}_{i,l} = (\mathbf{A}^{\top}\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^{\top}\mathbf{W}\mathbf{b}$, solved independently per tracker l. Fig. 4 demonstrates this.

Views whose motion direction is nearly collinear with the viewing ray (angle $< 25°$) are discarded: such configurations contribute little depth information and cause depth changes to appear as lateral motion. Geometrically, $\hat{\mu}^{3d}_{i,l}$ is the point minimizing perpendicular distance to all rays $\hat{\mathbf{d}}_{v,i,l}$.

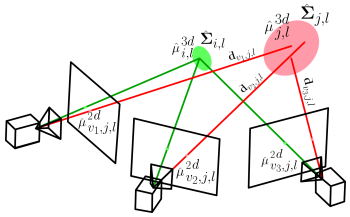

Fig. 4: We use gaussian-pixel matching to estimate 3D flow of each level

Covariance estimation. Each triangulation yields not just a position but a measure of geometric reliability. Under isotropic observation noise, the estimator covariance is:

$$\hat{\boldsymbol{\Sigma}}_{i,l} = \hat{\sigma}^2_{i,l} \cdot (\mathbf{A}^{\top}\mathbf{W}\mathbf{A})^{-1}, \qquad \hat{\sigma}^2_{i,l} = \frac{\|\mathbf{W}[\mathbf{A}\hat{\mu}^{3d}_{i,l} - \mathbf{b}]\|_2^2}{N_i - 3}, \tag{5}$$

where $N_i - 3$ is the residual degrees of freedom. Large eigenvalues of $\hat{\boldsymbol{\Sigma}}_{i,l}$ reflect poor geometry (near-parallel rays, few views, or inconsistent flow), while small eigenvalues reflect a reliable estimate. We pass $\hat{\boldsymbol{\Sigma}}_{i,l}$ directly as measurement uncertainty into the fusion below.

4.6 Variational Temporal Motion Fusion

We fuse the per-level triangulation estimates with a temporal prior through a single Kalman-style update, where each source contributes in proportion to its geometric precision.

Energy formulation. We seek the 3D position $\mu_{i,t}^{3d}$ of Gaussian i at frame t minimizing:

$$\mathcal{E}\big(\{\mu_{i,\tau}^{3d}\}_{\tau=1}^{t}\big) = \sum_{\tau=1}^{t} \|\mu_{i,\tau}^{3d} - \mu_{i,\tau-1}^{3d}\|_{\mathbf{Q}_\tau^{-1}}^2 + \sum_{l\in L} \|\mu_{i,t}^{3d} - \hat{\mu}_{i,l}^{3d}\|_{\hat{\boldsymbol{\Sigma}}_{i,l}^{-1}}^2, \tag{6}$$

where $\|\mathbf{v}\|_{\mathbf{M}}^2 = \mathbf{v}^\top \mathbf{M}\mathbf{v}$. The first sum penalizes temporal deviations weighted by inverse process noise $\mathbf{Q}_\tau^{-1}$; the second penalizes disagreement with each tracker weighted by its triangulation precision $\hat{\boldsymbol{\Sigma}}_{i,l}^{-1}$.

Closed-form minimizer. Minimizing $\mathcal{E}$ with respect to $\mu_{i,t}^{3d}$ yields the linear system:

$$\left[\mathbf{P}_{i,t|t-1}^{-1} + \sum_{l\in L} \hat{\boldsymbol{\Sigma}}_{i,l}^{-1}\right] \mu_{i,t}^{3d} = \mathbf{P}_{i,t|t-1}^{-1}\mu_{i,t-1}^{3d} + \sum_{l\in L} \hat{\boldsymbol{\Sigma}}_{i,l}^{-1}\hat{\mu}_{i,l}^{3d}, \tag{7}$$

where $\mathbf{P}_{i,t|t-1} = \mathbf{P}_{i,t-1} + \mathbf{Q}_t$ is the predicted covariance from the previous posterior. This is the information-form Kalman update; the posterior mean and covariance are:

$$\boldsymbol{\mu}_{i,t} = \boldsymbol{\Sigma}_{i,t}\left[\mathbf{P}_{i,t|t-1}^{-1}\mu_{i,t-1}^{3d} + \sum_{l\in L} \hat{\boldsymbol{\Sigma}}_{i,l}^{-1}\hat{\mu}_{i,l}^{3d}\right], \qquad \boldsymbol{\Sigma}_{i,t}^{-1} = \mathbf{P}_{i,t|t-1}^{-1} + \sum_{l\in L} \hat{\boldsymbol{\Sigma}}_{i,l}^{-1}. \tag{8}$$

Multi-resolution fusion and temporal propagation are therefore not independent design choices but two terms in the same objective, competing for influence in the precision accumulator $\boldsymbol{\Sigma}_{i,t}^{-1}$ purely on the basis of geometric reliability. Gaussians for which all levels fail to triangulate receive only the temporal term $\mathbf{P}_{i,t|t-1}^{-1}$, anchoring them to their previous position with covariance that grows with each unanswered frame.

We model dynamics as a random walk $\mu_{i,t}^{3d} = \mu_{i,t-1}^{3d} + w_t$, $w_t \sim \mathcal{N}(0,\, q_t^2\mathbf{I})$, where q_t is the median 3D displacement of successfully triangulated Gaussians at frame t, bootstrapped from the previous frame at $t = 1$. This ties process noise directly to observed scene dynamics: fast frames relax the temporal prior and cede influence to the tracker measurements; near-static frames tighten it.

Optimization. Following [17], we optimize dynamic and static layers in parallel using a hybrid renderer that jointly rasterizes Gaussians and the skybox. Dynamic Gaussians minimize a foreground-masked $\mathcal{L}_1$-SSIM loss over all parameters; background Gaussians update only their spherical harmonics to preserve geometric stability. Because the variational initialization places Gaussians near their correct positions, convergence requires very few iterations.

Uncertainty-driven densification. Rather than waiting for rendering error to identify under-reconstructed regions after many gradient steps, we split or clone Gaussians with high triangulation uncertainty before optimization begins, front-loading capacity where multi-view geometry is weakest. To the best of our knowledge, no prior streaming Gaussian method drives densification temporally (from geometric uncertainty) rather than rendering error.

5 Experiments

We evaluate FastFlowGS on two datasets testing complementary regimes. CMU-Panoptic [19] is a standard indoor multi-view benchmark, included to confirm that solving the large-displacement problem does not degrade general indoor performance. Monaco4D tests whether the approach holds when the assumptions underlying every prior method no longer do. For CMU-Panoptic, we select six subsequences from *161029_sports1* and evaluate at the native frame rate and under 5× subsampling to simulate larger inter-frame motion.

Baselines. We compare against six online reconstruction methods with public implementations: Dynamic-3DGS [34], 3DGStream [40], HiCoM [11], QUEEN [12], ReCon-GS [10], and TrackerSplat [54]. Instant Gaussian Stream [50] exceeded GPU memory on both datasets.

We additionally report **3DGS-Base**, a purposefully simple control that initializes each frame's dynamic Gaussians from the previous frame's reconstruction and finetunes with standard 3DGS, without predicting positions before optimization. Every other method, including FastFlowGS, explicitly estimates where Gaussians should move before optimization begins; 3DGS-Base isolates exactly what that prediction step is worth. On CMU-Panoptic, we report FastFlowGS which uses 300 iterations per frame, and FastFlowGS-faster which uses 200 iterations, trading a modest quality reduction for higher throughput.

Metrics. We report PSNR and VMAF averaged across all frames and views. Neither captures reconstruction quality at dynamic regions, so we additionally compute Masked PSNR (M-PSNR), restricting evaluation to dynamic foreground regions. A method that reconstructs static pixels accurately but smears fast-moving objects can post high full-image PSNR while failing at the actual task.

Most methods initialize from a high-quality first frame, inflating sequence-average PSNR while concealing temporal decay. We introduce **Δ-PSNR**, the difference between first- and last-frame PSNR, which is high for degrading methods,

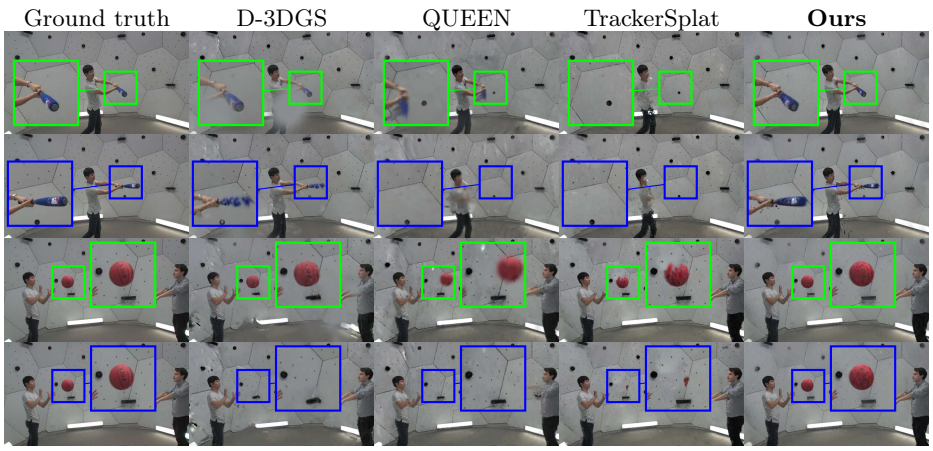

Fig. 5: Qualitative Results for test cameras of CMU-Panoptic dataset. To avoid overcrowding, we show results only on the top 3 quality methods. Insets mean the default frame rate and insets mean 5 frame skip

near zero for stable ones. For efficiency, we report mean per-frame training time using only the dynamic component of FastFlowGS, since static and dynamic branches run in parallel. VMAF Efficiency (VE = VMAF/Time) and PSNR Efficiency (PE = PSNR/Time) capture quality per unit of compute. All experiments run on an NVIDIA RTX A4500 at 960×540. Following [17,34,40,54], we report training time only, as preprocessing pipelines differ across methods. We provide an entire wall-clock time decomposition of each component (including preprocessing time) in Sec. 1.3 of the accompanying supplemental.

5.1 CMU-Panoptic

All methods run end-to-end with default pipelines on the selected subsequences.

FastFlowGS leads every baseline on every metric at both frame rates, but the more revealing result is temporal stability, not the absolute quality gap. Most

Table 1: Quantitative results on CMU-Panoptic at normal frame rate (left) and 5× frame skipping (right). *Time* is in seconds.

	Normal Frame Rate						5× Faster					
	VMAF ↑	PSNR ↑	ΔPSNR ↓	MPSNR ↑	Time ↓	VE ↑	VMAF ↑	PSNR ↑	ΔPSNR ↓	MPSNR ↑	Time ↓	VE ↑
D-3DGS	45.45	25.981	0.397	24.070	120.54	0.38	45.93	25.941	0.631	24.394	120.54	0.38
3DGStream	48.39	27.189	0.006	21.563	6.75	7.17	32.95	21.68	3.71	13.20	6.75	4.88
HiCoM	53.93	27.033	2.349	23.784	5.89	9.16	36.66	21.105	3.042	13.16	5.89	6.22
QUEEN	49.21	27.046	1.321	25.006	8.81	5.59	51.16	26.448	2.188	25.39	8.81	5.81
ReConGs	42.94	25.475	1.484	22.665	3.82	11.24	18.91	20.749	2.473	12.71	3.82	4.95
Trackersplat	46.24	24.719	3.825	22.536	14.14	3.27	44.36	24.364	2.928	19.791	14.14	3.14
Ours	60.71	27.893	0.133	26.141	4	15.18	50.66	27.571	0.192	27.19	4	12.67
Ours-faster	55.79	27.262	0.147	25.722	3	27.9	48.36	26.894	0.241	26.84	3	24.18

baselines accumulate more than 1 dB of drift over a sequence; FastFlowGS holds Δ-PSNR near zero at both frame rates (Tab. 1, Fig. 5).

The 5× subsampled setting reveals which motion models hold under stress. 3DGStream, HiCoM and ReCon-GS suffer the sharpest drops, because they all use (neural or hierarchical) hash grids which impose a finest-cell resolution that caps representable displacements. FastFlowGS retains highest PSNR, M-PSNR, and lowest Δ-PSNR under faster motion, confirming that multi-scale correspondence fusion does not require smooth inter-frame transitions to work. FastFlowGS-faster maintains the highest VE across both settings at 3s per frame.

Table 2: Results on Monaco4D under the full-quality protocol. All streaming baselines fail on this setting (Fig. 7), hence only 3DGS-Base is reported. *Time* is in seconds.

	Fairmont (5 cars)				Rascasse (1 car)				Main Straight (3 cars)			
	VMAF↑	PSNR ↑	MPSNR ↑	Time ↓	VMAF ↑	PSNR↑	MPSNR↑	Time↓	VMAF↑	PSNR↑	MPSNR↑	Time↓
Base	41.87	21.06	16.4	11.82	39.87	21.26	15.66	6.42	55.89	21.43	15.99	8.51
Ours	43.93	21.53	18.88	8.47	40.81	21.48	17.47	6.12	56.24	22.22	18.96	7.11

5.2 Monaco4D

Monaco4D is not simply a harder version of CMU-Panoptic, because it tests a regime where the assumptions underlying existing methods break rather than bend. Both evaluation protocols below share the same scene factorization: the static background uses hierarchical 3DGS [32], the skybox is rendered from a cubemap [51], and the dynamic foreground is optimized per frame. This decomposition is necessary because end-to-end optimization at full resolution exceeds the GPU memory of every method we evaluate. All methods use the same static and sky layers, so the comparison isolates each method's dynamic reconstruction.

Full-quality evaluation. Under this protocol the static background uses the full hierarchical reconstruction at roughly 8 million Gaussians, exceeding the memory budget of all streaming baselines; the only valid comparison is therefore against

Fig. 6: Qualitative Results for test cameras of Monaco4D. Insets depict the renders from our 3DGS baseline and insets are from FastFlowGS.

3DGS-Base. That this comparison reduces to a single baseline is not a limitation of the experimental design rather, the central empirical finding of this section, and we return to it directly with the reduced-memory experiment below.

We evaluate on three sequences spanning increasing scene complexity: Fairmont puts five mutually occluding cars through a tight hairpin, Main Straight runs three cars at maximum speed, Rascasse, the simplest, has a single car.

FastFlowGS outperforms 3DGS-Base on every sequence and metric (Tab. 2). The gains are correlated with scene complexity, largest on Fairmont and smallest on Rascasse. Full-image PSNR understates the difference because the static background dominates the per-pixel average. 3DGS-Base produces visibly smeared car surfaces with lost high-frequency detail that appear clearly in Fig. 6 but are diluted by full-image metrics.

Why existing streaming methods fail. The failure of streaming baselines on Monaco4D is not a matter of degree. Trained on masked dynamic regions under the full-quality protocol, all five methods lose the majority of their Gaussians by frame 2 and produce empty reconstructions by frame 5. Three conditions, absent in combination from every prior streaming benchmark, compound to produce this outcome:

1. Inter-frame displacements are an order of magnitude beyond what any existing method was designed or evaluated for.
2. Cars routinely leave and re-enter individual camera frustums between consecutive frames, breaking the assumption of continuous multi-view coverage that underpins every baseline.
3. At most 7–8 cameras observe the dynamic region at once, providing far sparser per-object supervision than any indoor benchmark.

Methods without densification, D-3DGS and TrackerSplat, degrade monotonically because Gaussians displaced beyond the dynamic mask lose their gradient signal and are never recovered. Methods with densification, 3DGStream, HiCoM, ReCon-GS, and QUEEN, eventually regenerate an approximate shape, but the initial estimate collapses so completely that recovery is indistinguishable from restarting 3DGS from scratch on each frame (Fig. 7).

Fig. 7: Renders from a training view at frame 2. All streaming baselines initialized on frame 1 lose their Gaussians within one step. Large displacements push primitives outside the dynamic mask, severing gradient flow before optimization can recover.

Table 3: Quantitative results with reduced-memory background, per sequence. *Time* is in seconds. QUEEN ran out of memory on Main Straight (–). Full results including PSNR and MPSNR are in the supplemental. (Tab. 3)

	Fairmont			Main Straight			Rascasse			Tunnel			Uphill			Pool		
Method	VMAF	Time	VE	VMAF	Time	VE	VMAF	Time	VE	VMAF	Time	VE	VMAF	Time	VE	VMAF	Time	VE
D-3DGS	40.47	738	0.05	35.83	724	0.05	38.73	729	0.05	32.11	732	0.04	41.17	740	0.06	34.36	727	0.05
HiCoM	0.89	27	0.03	1.09	24	0.05	2.01	25	0.08	0.25	24	0.01	0.97	29	0.03	1.16	26	0.04
QUEEN	5.39	61	0.09	–	–	–	12.67	23	0.58	0.39	69	0.01	0.22	70	0	38.75	23	1.68
ReConGs	0.71	48	0.01	1.21	48	0.03	1.66	35	0.05	0.36	39	0.01	0.89	39	0.02	1.24	25	0.05
Trackersplat	2.72	72	0.04	0.78	95	0.01	9.67	70	0.14	7.15	94	0.08	1.59	60	0.03	6.77	58	0.12
FastFlowGS	46.89	25	1.88	50.1	21	2.39	53.84	23	2.34	48.9	19	2.57	55.84	27	2.07	51.87	22	2.36

Reduced-memory evaluation. To test whether this failure is intrinsic to the displacement regime or merely a consequence of memory constraints, we replace the hierarchical background with a standard 3DGS at one-eighth the point count, roughly 1 million Gaussians, bringing every method within budget. Per-timestep iterations are increased to allow convergence, so FastFlowGS numbers in Tab. 3 differ slightly from the full-quality protocol.

The answer is unambiguous (Tab. 3). Despite operating against a background eight times denser than any baseline, FastFlowGS achieves the highest VE on every sequence. The gap is not primarily about raw timing: on Fairmont, HiCoM finishes within two seconds of FastFlowGS (27 s vs. 25 s), yet produces VMAF 0.89 against FastFlowGS's 46.89. D-3DGS is the only baseline that achieves recognizable reconstruction quality (VMAF 32–41 across sequences), but at 720–740 seconds per frame it is two orders of magnitude slower than FastFlowGS. QUEEN shows its best result on Pool, where its VE of 1.68 is the closest any baseline comes to FastFlowGS's 2.36, but its VMAF of 38.75 trails by 13 points and it collapses on every other sequence. HiCoM, ReCon-GS, and TrackerSplat fall below VMAF 10 on most sequences. Relaxing the memory constraint does not resolve the failure, which confirms that the problem is a fundamental mismatch between the displacement regime Monaco4D introduces and the motion models these methods rely on. Qualitative results in Fig. 1 in supplemental.

5.3 Ablations

Tab. 4 and Fig. 8 show that every component contributes, and the gains compound rather than overlap.

Sparse and dense correspondences alone reach similar quality on CMU-Panoptic (M-PSNR 25.31 vs. 25.29), with dense converging modestly faster. On Monaco4D, the advantage of dense correspondences is clearer (VMAF 40.76 vs. 39.33), consistent with the greater prevalence of textureless surfaces and specular bodywork that makes sparse feature matching unreliable outdoors. Fusing both outperforms either in isolation on every metric, confirming that the two signals cover different failure modes rather than providing redundant coverage.

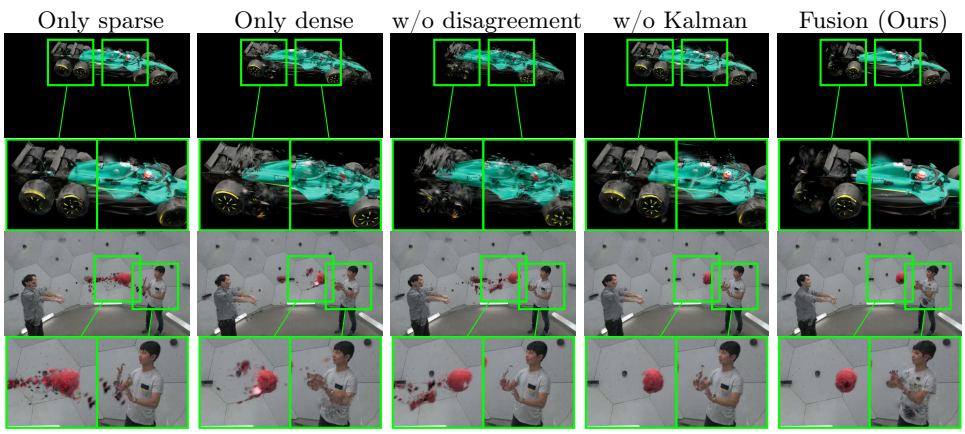

Fig. 8: Ablation study on Monaco4D (top) and CMU-Panoptic (bottom). We additionally report #iters: the number of iterations to reach 24+ PSNR on CMU-Panoptic and 18.5+ PSNR on Monaco4D.

The disagreement filter's primary contribution is to convergence rather than final quality, removing 40 iterations on CMU-Panoptic and 80 on Monaco4D before the quality threshold is reached. The larger effect outdoors is consistent with the higher rate of flow failures on Monaco4D, where filtering unreliable correspondences before they enter the optimizer has more noise to suppress.

The Kalman temporal prior delivers the single largest improvement across all components and both datasets: VMAF rises by 2.55 points on CMU-Panoptic and 2.76 on Monaco4D, and convergence drops from 260 to 200 iterations and from 760 to 700 respectively. We attribute this to the prior constraining Gaussian positions to a temporally smooth trajectory, reducing the search space the optimizer must explore at each new frame.

Component interactions. The disagreement filter operates on the fused correspondence set, which already combines sparse and dense coverage; applied to either signal alone, it would identify fewer cross-signal disagreements and suppress less noise. The Kalman prior then acts on a well-filtered set, and its gain is largest precisely because upstream filtering has removed the noise that would otherwise corrupt the temporal smoothness assumption.

6 Conclusion

The central finding of this work is that initialization quality, not optimization budget, is the bottleneck in streaming Gaussian reconstruction. By fusing multi-resolution correspondences with a Kalman temporal prior yields initializations that converge faster and remain stable over long sequences. FastFlowGS instantiates this principle and outperforms all streaming baselines on CMU-Panoptic. On Monaco4D, the first outdoor multi-view dataset in the large-displacement regime released publicly alongside this work, it is the only method to produce

Table 4: Per-component ablation. *#iters* is iterations to reach 24+ PSNR on CMU-Panoptic and 18.5+ PSNR on Monaco4D (averaged over training views).

				CMU Panoptic (Football)			Monaco4D (Fairmont)		
Sparse	Dense	Disag.	Kalman	VMAF↑	MPSNR↑	#iters↓	VMAF↑	MPSNR↑	#iters↓
✓	✗	✗	✗	46.14	25.31	350	39.33	18.65	1000
✗	✓	✗	✗	45.32	25.29	320	40.76	18.64	940
✓	✓	✗	✗	46.49	25.37	300	40.84	18.71	840
✓	✓	✓	✗	46.82	25.43	260	41.17	18.76	760
✓	✓	✓	✓	49.37	25.58	200	43.93	18.88	700

coherent reconstructions where all existing approaches fail entirely. We hope Monaco4D shifts the community's attention toward the displacement and sparsity regimes that outdoor broadcast applications actually require, rather than the smooth, dense-view conditions that current benchmarks favor.

7 Limitations and Failure Cases

The motion model updates positions but not rotations or scales, so deforming or rapidly rotating objects must recover shape from scratch, eroding the convergence advantage that good initialization provides. We also acknowledge that correspondence quality degrades on textureless surfaces, distant objects, and depth-dominant motion, where disagreement scoring helps but cannot fully compensate. Variational fusion treats tracker levels as independent, making the fused covariance $\boldsymbol{\Sigma}_{i,t}$ overconfident when levels share image evidence and reducing the optimizer's ability to correct under tight iteration budgets. Both FastFlowGS and Monaco4D assume a static background, so moving spectators or trackside elements produce localized ghosting, and the method has no recovery path when segmentation masks fail across multiple views simultaneously. Monaco4D is synthetic, and while FastFlowGS transfers to real capture on CMU-Panoptic, performance on real outdoor footage at F1-scale displacements remains uncharacterized. Preprocessing is dominated by the point tracker at 727 s per frame and is excluded from timing comparisons following standard practice [17, 34, 40, 54], with the full breakdown in Sec. 1.3 of the supplemental.

Acknowledgments We thank Javier Sevilla, Juanfran Ramírez, Arturo Javier Loza, and Daniel Fernandez from BionicApe for their guidance and advice during dataset creation, and Aviral Chharia for helpful suggestions during the review process.

References

1. 2026 formula 1 technical regulations, https://www.fia.com/sites/default/files/fia_2026_formula_1_technical_regulations_issue_8_-_2024-06-24.pdf
2. Hawkeye, howpublished = https://www.hawkeyeinnovations.com/,
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
4. 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)
5. Chen, Z., Qin, M., Yuan, T., Liu, Z., Zhao, H.: Long3r: Long sequence streaming 3d reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5273–5284 (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. Epic Games: Unreal engine, https://www.unrealengine.com
8. Fang, G., Wang, B.: Mini-splatting: Representing scenes with a constrained number of gaussians. In: European conference on computer vision. pp. 165–181. Springer (2024)
9. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12479–12488 (2023)
10. Fu, J., Gao, Q., Wen, C., Wu, Y., Ma, S., Zhang, J., Zhang, J.: Recon-gs: Continuum-preserved gaussian streaming for fast and compact reconstruction of dynamic scenes. arXiv preprint arXiv:2509.24325 (2025)
11. Gao, Q., Meng, J., Wen, C., Chen, J., Zhang, J.: Hicom: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting. Advances in Neural Information Processing Systems **37**, 80609–80633 (2024)
12. Girish, S., Li, T., Mazumdar, A., Shrivastava, A., De Mello, S., et al.: Queen: Quantized efficient encoding of dynamic gaussians for streaming free-viewpoint videos. Advances in Neural Information Processing Systems **37**, 43435–43467 (2024)
13. Gossard, T., Ziegler, A., Zell, A.: Tt3d: Table tennis 3d reconstruction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5821–5831 (2025)
14. Guo, J., Deng, N., Li, X., Bai, Y., Shi, B., Wang, C., Ding, C., Wang, D., Li, Y.: Streetsurf: Extending multi-view implicit surface reconstruction to street views. arXiv preprint arXiv:2306.04988 (2023)
15. Han, J., An, H., Jung, J., Narihira, T., Seo, J., Fukuda, K., Kim, C., Hong, S., Mitsufuji, Y., Kim, S.: Dˆ 2ust3r: Enhancing 3d reconstruction with 4d pointmaps for dynamic scenes. arXiv preprint arXiv:2504.06264 (2025)
16. Hong, Z.: Free-viewpoint video of outdoor sports using a flying camera. In: European Conference on Computer Vision. pp. 178–195. Springer (2025)
17. Huang, J., Subhajyoti Mallick, S., Amat, A., Ruiz Olle, M., Mosella-Montoro, A., Kerbl, B., Vicente Carrasco, F., De la Torre, F.: Echoes of the coliseum: Towards

3d live streaming of sports events. ACM Trans. Graph. **44**(4) (Jul 2025). https://doi.org/10.1145/3731214, https://doi.org/10.1145/3731214
18. Huang, Y.C., Liao, I.N., Chen, C.H., İk, T.U., Peng, W.C.: Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications. In: 2019 16th IEEE international conference on advanced video and signal based surveillance (AVSS). pp. 1–8. IEEE (2019)
19. Joo, H., Simon, T., Li, X., Liu, H., Tan, L., Gui, L., Banerjee, S., Godisart, T.S., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social interaction capture. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
20. 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)
21. Karhade, J., Keetha, N., Zhang, Y., Gupta, T., Sharma, A., Scherer, S., Ramanan, D.: Any4d: Unified feed-forward metric 4d reconstruction. arXiv preprint arXiv:2512.10935 (2025)
22. Kästingschäfer, M., Gieruc, T., Bernhard, S., Campbell, D., Insafutdinov, E., Najafli, E., Brox, T.: Seed4d: A synthetic ego-exo dynamic 4d data generator, driving dataset and benchmark. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 7752–7764. IEEE (2025)
23. Kerbl, B., Meuleman, A., Kopanas, G., Wimmer, M., Lanvin, A., Drettakis, G.: A hierarchical 3d gaussian representation for real-time rendering of very large datasets. ACM Transactions On Graphics (TOG) **43**(4), 1–15 (2024)
24. Khafizov, R., Komarichev, A., Rakhimov, R., Wonka, P., Burnaev, E.: G-cut3r: Guided 3d reconstruction with camera and depth prior integration. arXiv preprint arXiv:2508.11379 (2025)
25. Le Moing, G., Ponce, J., Schmid, C.: Dense optical tracking: Connecting the dots. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19187–19197 (2024)
26. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European conference on computer vision. pp. 71–91. Springer (2024)
27. Li, H., Li, S., Gao, X., Batuer, A., Yu, L., Liao, Y.: Gifstream: 4d gaussian-based immersive video with feature stream. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21761–21770 (2025)
28. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3d video synthesis from multi-view video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5521–5531 (2022)
29. Li, Z., Chen, Z., Li, Z., Xu, Y.: Spacetime gaussian feature splatting for real-time dynamic view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8508–8520 (2024)
30. Liang, Y., Khan, N., Li, Z., Nguyen-Phuoc, T.H., Lanman, D., Tompkin, J., Xiao, L.: Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. In: Proceedings of the Winter Conference on Applications of Computer Vision. pp. 2642–2652 (2025)
31. Liao, Y., Xie, J., Geiger, A.: Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(3), 3292–3310 (2022)

32. Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y., Yang, W.: Vastgaussian: Vast 3d gaussians for large scene reconstruction. In: CVPR (2024)
33. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: LightGlue: Local Feature Matching at Light Speed. In: ICCV (2023)
34. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: 3DV (2024)
35. 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. SA '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3680528.3687694, https://doi.org/10.1145/3680528.3687694
36. Oh, S., Lee, Y., Jeon, H., Park, E.: Hybrid 3d-4d gaussian splatting for fast dynamic scene representation. arXiv preprint arXiv:2505.13215 (2025)
37. 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 (2020)
38. Rander, P., Narayanan, P.J., Kanade, T.: Virtualized reality: constructing time-varying virtual worlds from real world events. In: Proceedings of the 8th Conference on Visualization '97. p. 277–ff. VIS '97, IEEE Computer Society Press, Washington, DC, USA (1997)
39. Shishido, H., Kameda, Y., Ohta, Y.: Trajectory estimation of a fast and anomalously moving badminton shuttle (2014), https://api.semanticscholar.org/CorpusID:18162110
40. Sun, J., Jiao, H., Li, G., Zhang, Z., Zhao, L., Xing, W.: 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20675–20685 (2024)
41. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
42. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. pp. 402–419. Springer (2020)
43. Wang, C., MacDonald, L.E., Jeni, L.A., Lucey, S.: Flow supervision for deformable nerf. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21128–21137 (2023)
44. Wang, L., Yao, K., Guo, C., Zhang, Z., Hu, Q., Yu, J., Xu, L., Wu, M.: Videorf: Rendering dynamic radiance fields as 2d feature video streams. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 470–481 (2024)
45. Wang, L., Zhang, J., Liu, X., Zhao, F., Zhang, Y., Zhang, Y., Wu, M., Yu, J., Xu, L.: Fourier plenoctrees for dynamic radiance field rendering in real-time. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13524–13534 (2022)
46. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10510–10522 (2025)
47. 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)

48. 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 (CVPR). pp. 20310–20320 (June 2024)
49. Xu, Z., Li, Z., Dong, Z., Zhou, X., Newcombe, R., Lv, Z.: 4dgt: Learning a 4d gaussian transformer using real-world monocular videos (2025)
50. Yan, J., Peng, R., Wang, Z., Tang, L., Yang, J., Liang, J., Wu, J., Wang, R.: Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16520–16531 (2025)
51. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians for modeling dynamic urban scenes. In: ECCV (2024)
52. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., Wang, Y.: Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. arXiv preprint arXiv:2311.02077 (2023)
53. 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)
54. Yin, D., Ding, I., Jin, Y., Shi, J., Liu, J.: Trackersplat: Exploiting point tracking for fast and robust dynamic 3d gaussians reconstruction. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. SA Conference Papers '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3757377.3763829, https://doi.org/10.1145/3757377.3763829
55. Zhang, C., Le Moing, G., Koppula, S., Rocco, I., Momeni, L., Xie, J., Sun, S., Sukthankar, R., Barral, J.K., Hadsell, R., Ghahramani, Z., Zisserman, A., Zhang, J., Sajjadi, M.S.M.: Efficiently reconstructing dynamic scenes one d4rt at a time. In: CVPR (2026)
56. Zhang, C., Moing, G.L., Koppula, S., Rocco, I., Momeni, L., Xie, J., Sun, S., Sukthankar, R., Barral, J.K., Hadsell, R., et al.: Efficiently reconstructing dynamic scenes one d4rt at a time. arXiv preprint arXiv:2512.08924 (2025)
57. 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. In: Yue, Y., Garg, A., Peng, N., Sha, F., Yu, R. (eds.) International Conference on Learning Representations. vol. 2025, pp. 82863–82886 (2025), https://proceedings.iclr.cc/paper_files/paper/2025/file/ce1710ff5b66f89b0d707bf1ec2850bf-Paper-Conference.pdf
58. Zhu, R., Liang, Y., Chang, H., Deng, J., Lu, J., Yang, W., Zhang, T., Zhang, Y.: Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. Advances in Neural Information Processing Systems **37**, 101790–101817 (2024)