FILT3R: Latent State Adaptive Kalman Filter for Streaming 3D Reconstruction

Seonghyun Jin[1] and Jong Chul Ye[1]*

Kim Jaechul Graduate School of AI, KAIST, Seoul, South Korea
jinotter3@kaist.ac.kr, jong.ye@kaist.ac.kr
https://github.com/jinotter3/FILT3R

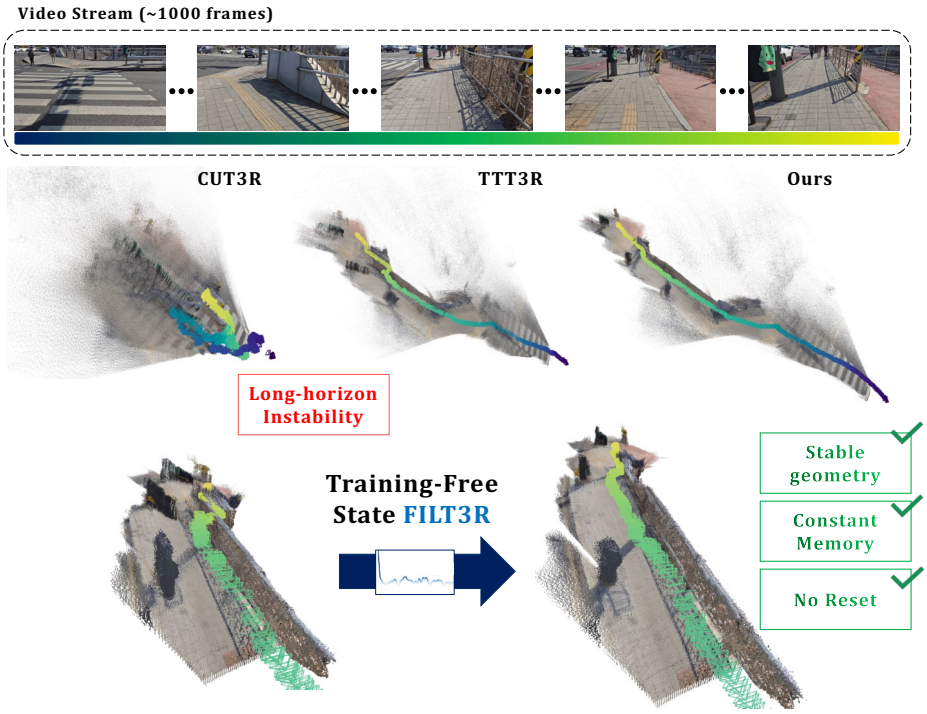


Fig. 1: FILT3R enables stable long-horizon streaming 3D reconstruction without resets. Top: dense input frames from a long video stream. Middle: final reconstructions from the same frozen streaming backbone under different latent-state update rules. Uniform overwrite (CUT3R) and heuristic gating (TTT3R) accumulate drift and/or forget previously integrated evidence, leading to long-horizon instability and degraded geometry. Bottom: a zoom-in around frames beyond 600 highlights that FILT3R preserves stable geometry over long rollouts. Trajectory colors indicate time.

Abstract. Streaming 3D reconstruction maintains a persistent latent state that is updated online from incoming frames, enabling constant-memory inference. A key failure mode is the state update rule: aggressive

* Corresponding author.

overwrites forget useful history, while conservative updates fail to track new evidence, and both behaviors become unstable beyond the training horizon. To address this challenge, we propose **FILT3R**, a training-free latent filtering layer that casts recurrent state updates as stochastic state estimation in token space. FILT3R maintains a per-token variance and computes a Kalman-style gain that adaptively balances memory retention against new observations. Process noise – governing how much the latent state is expected to change between frames – is estimated online from EMA-normalized temporal drift of candidate tokens. Using extensive experiments, we demonstrate that FILT3R yields an interpretable, plug-in update rule that generalizes common overwrite and gating policies as special cases. Specifically, we show that gains shrink in stable regimes as uncertainty contracts with accumulated evidence, and rise when genuine scene change increases process uncertainty, improving long-horizon stability for depth, pose, and 3D reconstruction, compared to the existing methods.

Keywords: Streaming 3D Reconstruction · State Estimation · Kalman Filtering · Length Generalization

1 Introduction

Streaming 3D reconstruction aims to predict geometry and camera motion from an image stream while using constant memory and low latency. Recent recurrent architectures achieve this by maintaining a compact set of persistent latent tokens, which distill the scene's context and evolve dynamically as new frames arrive [33]. In practice, performance often degrades when the sequence length exceeds the training horizon: small update biases accumulate into drift, and the latent state gradually loses information that is no longer supported by the most recent frame. Since a new candidate latent state should be estimated from the current frame and the previous state, the system must then decide how much to trust the candidate (fast adaptation) versus the previous state (long-term consistency).

Existing streaming update rules are mostly manually engineered to manage this trade-off. Uniform overwrites (*e.g.*, CUT3R [33]) always trust the new candidate, which can lead to catastrophic forgetting. While training-free gates based on attention statistics (*e.g.*, TTT3R [4]) improve stability, they remain largely heuristic; they scale updates without an explicit representation of uncertainty or a principled mechanism for adaptation as confidence increases. Mathematically, existing streaming updates can be written as $\mathbf{s}_t = (1 - \boldsymbol{\beta}_t) \odot \mathbf{s}_{t-1} + \boldsymbol{\beta}_t \odot \tilde{\mathbf{s}}_t$, where $\boldsymbol{\beta}_t \in [0, 1]^N$ determines how quickly past evidence is forgotten. Uniform overwrite ($\boldsymbol{\beta}_t = \mathbf{1}$) yields a one-step memory, while fixed-gain gates impose a constant forgetting half-life regardless of stream stability.

Inspired by this formulation, we argue that the recurrent state in streaming reconstruction is naturally a *belief state*, and that the candidate state produced by the decoder is a noisy *measurement* of that belief. This viewpoint leads directly to our novel adaptive filtering formulation, which we call **FILT3R**. In

particular, FILT3R formulates the state update as a Kalman-style interpolation whose gain is governed by two uncertainties: (i) *process noise*—how much the underlying scene latent is expected to change between frames—and (ii) *measurement noise*—how reliable the decoder's current prediction is. Specifically, FILT3R adopts an adaptive Kalman-style filtering (AKF) framework [15] with a fixed measurement model. Adaptivity is centered on the process model, allowing Kalman-style gains to attenuate as tokens gain confidence during stable periods – thereby extending the effective memory horizon – and increase in response to process noise that indicates a genuine scene change. More specifically, FILT3R uses a *fixed* scalar measurement noise r and estimates only the *process noise* adaptively from temporal drift in the latent space. This formulation not only reduces computational complexity but also circumvents a critical failure mode, as validated by our empirical findings.

Our contribution can be summarized as follows:

(i) We recast streaming 3D reconstruction as stochastic state estimation in latent token space, and introduce **FILT3R**, an adaptive latent filtering layer for streaming 3D reconstruction.
(ii) FILT3R is a training-free, lightweight, and interpretable plug-and-play filtering approach with adaptive process noise estimated from internal model signals and fixed measurement noise, following the classical AKF paradigm.
(iii) We provide extensive evaluation across short- and long-horizon streaming depth, pose, and reconstruction settings, and analyze how uncertainty-aware gains improve length generalization. The results confirm that FILT3R consistently outperforms prior streaming baselines and improves long-horizon stability across metrics.

2 Related Work

Streaming and recurrent 3D reconstruction. Classical and learning-based 3D streaming pipelines have progressed from short-horizon depth/pose models to online recurrent reconstruction. Representative geometric estimators include RAFT-style correlation, DeepV2D, and DROID-SLAM [25–27]. For online scene understanding, methods such as iMAP, NICE-SLAM, Co-SLAM, Point-SLAM, and MASt3R-SLAM maintain a persistent latent map updated as frames arrive [16, 18, 23, 31, 41]. Recent foundation-style 3D models further improve feed-forward reconstruction quality, from DUSt3R/MASt3R to larger multi-view architectures including VGGT and its follow-ups [5,11,13,20,32,34,35,37]. Streaming variants such as Spann3R, CUT3R, SLAM3R, Point3R, Stream3R, StreamVGGT, and WinT3R explicitly target long-horizon online inference under bounded memory [10, 12, 14, 30, 33, 36, 42].

A central design choice is the state update rule in recurrent memories. CUT3R uses uniform overwrite [33], while recent training-free methods introduce adaptive gating signals from attention or motion cues, including TTT3R, TTSA3R, and MUT3R [4, 19, 40]. In parallel, test-time adaptation methods update model parameters online, *e.g.*, test-time training and entropy minimization [24, 29].

In streaming 3D reconstruction, errors compound with rollout length, causing drift and forgetting beyond the training horizon. This failure mode is visible in overwrite-based and heuristic-gated recurrent updates [4, 33, 40]. Our goal is to address this by explicitly propagating uncertainty so gains decay in stable periods and rise only when estimated process uncertainty indicates genuine scene change.

Uncertainty estimation and filtering. Kalman-style filtering provides a principled framework for sequential estimation under linear-Gaussian assumptions [8]. Adaptive Kalman-style filtering estimates noise covariances online when dynamics are non-stationary [15]. Neural and differentiable variants include Deep Kalman Filters and Backprop KF [7, 9]. We adopt this perspective for latent token memories, using adaptive process noise and propagated uncertainty to derive per-token gains while keeping measurement noise fixed.

3 FILT3R

3.1 Problem formulation

We consider a recurrent streaming reconstruction model that maintains persistent state tokens $\mathbf{s}_t \in \mathbb{R}^{N\times D}$ and, given an incoming frame at time t, produces a candidate state token $\tilde{\mathbf{s}}_t \in \mathbb{R}^{N\times D}$ using a decoder that attends to image tokens and the previous state [33]. Here, N and D refer to the number of tokens and token dimension, respectively.

Specifically, we interpret $\mathbf{s}_t$ as a *belief* over the scene latent at time t and $\tilde{\mathbf{s}}_t$ as a noisy *measurement* of that belief produced by the decoder. This suggests modeling the dynamics with a stochastic state-space model in token space:

$$\mathbf{s}_t = \mathbf{s}_{t-1} + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t), \tag{1}$$

$$\tilde{\mathbf{s}}_t = \mathbf{s}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{2}$$

where $\mathbf{Q}_t$ models uncertainty in latent evolution (process noise) and $\mathbf{R}$ models uncertainty in the decoder's prediction (measurement noise).

Streaming video is non-stationary: the scene can change rapidly during motion and remain static during pauses. We therefore adopt an *adaptive Kalman-style filtering* (AKF) approach [15], estimating process noise $\mathbf{Q}_t$ online from the model's internal representations.

A key design choice in FILT3R is to keep measurement noise *fixed* using a single scalar shared across tokens, *i.e.* $\mathbf{R} = r\,\mathbf{I}$. The rationale is that the decoder is a frozen pretrained network: its prediction quality is determined by its architecture and training, not by the current scene dynamics. In classical AKF, this corresponds to the standard assumption that the sensor noise is known [15].

In contrast, we found that making both $\mathbf{Q}_t$ and $\mathbf{R}_t$ adaptive introduces the risk of correlated noise estimates: scene transitions simultaneously increase temporal drift (which feeds $\mathbf{Q}_t$) and attention novelty (which would feed $\mathbf{R}_t$). When both spike together, the gain $\mathbf{k}_t$ can overshoot, producing the same instability that heuristic gates suffer from. A fixed r in our FILT3R provides a stable anchor, and all adaptivity enters through the process noise.

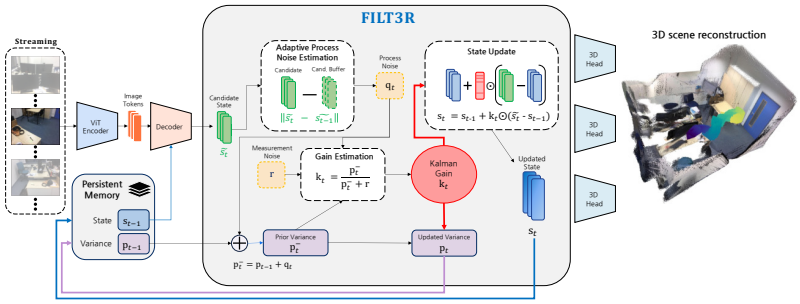


Fig. 2: FILT3R overview. The persistent memory $\mathbf{s}_{t-1}$ is fused with the decoder's candidate $\tilde{\mathbf{s}}_t$ via a token-wise Kalman-style gain $\mathbf{k}_t$. Process noise $\mathbf{q}_t$ is estimated adaptively from EMA-normalized temporal drift, while measurement noise r is a scalar hyperparameter shared across tokens. Variance $\mathbf{p}_t$ is propagated across time steps, enabling gains that naturally shrink in stable regimes and increase during scene change.

3.2 Update rule

We assume a diagonal per-token variance $\mathbf{p}_t \in \mathbb{R}^N$ (one scalar per token), approximating each token's covariance as isotropic. Equivalently, we use diagonal covariance approximations $\mathbf{Q}_t = \mathrm{diag}(\mathbf{q}_t)$ and $\mathbf{P}_t = \mathrm{diag}(\mathbf{p}_t)$, which yields elementwise Kalman updates. Then, the variance prediction follows the standard Kalman prediction step:

$$\mathbf{p}_t^- = \mathbf{p}_{t-1} + \mathbf{q}_t, \tag{3}$$

where $\mathbf{q}_t \in \mathbb{R}^N$ is the per-token process noise estimated online (Sec. 3.2), and the Kalman-style gain is given by

$$\mathbf{k}_t = \frac{\mathbf{p}_t^-}{\mathbf{p}_t^- + r}, \tag{4}$$

Consequently, the state update according to AKF is a gain-weighted interpolation:

$$\mathbf{s}_t = \mathbf{s}_{t-1} + \mathbf{k}_t \odot (\tilde{\mathbf{s}}_t - \mathbf{s}_{t-1}). \tag{5}$$

and the variance is updated using the scalar Joseph form:

$$\mathbf{p}_t = (\mathbf{1} - \mathbf{k}_t)^2 \odot \mathbf{p}_t^- + r\mathbf{k}_t^2. \tag{6}$$

where $\odot$ refers to the Hadamard product, and $\mathbf{k}_t^2 = \mathbf{k}_t \odot \mathbf{k}_t$.

Uncertainty modeling of latent evolution. Given the update rules for $\mathbf{s}_t, \mathbf{p}_t$, AKF formulation requires the process noise variance that captures how much the scene latent is expected to change between frames. In a static scene, $\mathbf{q}_t$ should be small; during rapid motion or scene transitions, $\mathbf{q}_t$ should be large. FILT3R estimates this from the temporal drift of consecutive candidate states, using a per-stream EMA baseline to calibrate the scale.

Specifically, we model the i-th token process noise through a sigmoid gate:

$$q_{t,i} = q_{\min} + (q_{\max} - q_{\min}) \cdot \sigma\big(\alpha_q(g_{t,i} - \tau_q)\big), \tag{7}$$

where $q_{\min}$, $q_{\max}$ bound the process noise, α_q controls the sharpness of the transition, and τ_q is the midpoint, and $g_{t,i}$ refers to the normalized drift score with respect to the running EMA baseline of the stream-level mean drift $\hat{\Delta}_t$:

$$g_{t,i} = \Delta_{t,i}/\hat{\Delta}_t \tag{8}$$

$$\hat{\Delta}_t = (1 - \lambda_\Delta)\,\hat{\Delta}_{t-1} + \lambda_\Delta \bar{\Delta}_t, \tag{9}$$

where λ_Δ is the EMA rate and

$$\bar{\Delta}_t := \sum_i \Delta_{t,i}/N, \quad \text{where} \quad \Delta_{t,i} = \|\tilde{\mathbf{s}}_{t,i} - \tilde{\mathbf{s}}_{t-1,i}\|_2. \tag{10}$$

The normalized drift score measures whether the current drift is high or low *relative to the stream's running baseline*.

Intuitively, when the drift is typical (near the EMA baseline), the sigmoid is near its midpoint and $q_{t,i}$ takes a moderate value; when the drift is unusually large, $q_{t,i}$ approaches $q_{\max}$, signaling rapid scene change; when the drift is small, $q_{t,i}$ approaches $q_{\min}$, indicating stability.

Implementation issue. For numerical stability, we add small ϵ to the denominators of (4) and (8) to avoid the division by a near zero value. The gain in (4) is additionally clamped to $[k_{\min}, k_{\max}]$ to prevent complete freezing or full overwrite. Similarly, the EMA baseline is clamped to a floor Δ_{floor} to avoid division instability.

The overhead of our AKF formulation is minimal: one N-dimensional vector of token variances, one per-stream EMA scalar for normalizing temporal drift, and one $N \times D$ buffer for the previous candidate.

3.3 Filtering as adaptive token retention

The Kalman-style gain $\mathbf{k}_t$ in Eq. (5) acts as a per-token update coefficient $\boldsymbol{\beta}_t := \mathbf{k}_t \in [0,1]^N$, yielding an exponential-smoothing form $\mathbf{s}_t = (1 - \boldsymbol{\beta}_t) \odot \mathbf{s}_{t-1} + \boldsymbol{\beta}_t \odot \tilde{\mathbf{s}}_t$. Unlike overwrite ($\boldsymbol{\beta}_t \equiv \mathbf{1}$) or fixed/heuristic gates, FILT3R propagates uncertainty (Eqs. (3) and (6)), so gains *naturally shrink* as the state becomes confident in stable regimes, extending the effective memory horizon, while still producing gain increases when inferred process noise favors adaptation.

More specifically, the Kalman-style gain (Eq. (4)) is entirely driven by the ratio $\mathbf{p}_t^-/(\mathbf{p}_t^- + r)$. In stable regimes, $\mathbf{q}_t$ stays near $q_{\min}$, so $\mathbf{p}_t^-$ gradually decreases as the Joseph update (Eq. (6)) shrinks the variance, causing $\mathbf{k}_t$ to decay and the effective memory horizon to grow. During scene transitions, $\mathbf{q}_t$ spikes, $\mathbf{p}_t^-$ increases, and $\mathbf{k}_t$ rises, allowing the filter to rapidly integrate new evidence. With $\mathbf{q}_t$ driving $\mathbf{p}_t^-$, the gain in Eq. (4) increases under large drift and decreases as uncertainty contracts in stable regimes.

This formulation clarifies the relationship to prior update rules:

- *Uniform overwrite* (CUT3R): $\boldsymbol{\beta}_t \equiv \mathbf{1}$. Equivalent to $r = 0$ (infinite trust in the measurement).
- *Heuristic gates* (TTT3R): $\boldsymbol{\beta}_t$ computed from instantaneous attention/change statistics without uncertainty propagation. Equivalent to resetting variance to a constant each step.
- *Fixed interpolation*: $\boldsymbol{\beta}_t \equiv \beta$. A single point on the steady-state q/r curve (see Appendix A.1).

In fact, in FILT3R, $\boldsymbol{\beta}_t$ is derived via recursive uncertainty updates (Eqs. (3), (4) and (6)). This yields Kalman-style gains that decay at a rate of $\mathcal{O}(1/t)$ in static scenes (see Appendix A.1, Prop. 1) but spike in response to process noise signaling a scene change. To support this mechanism, Appendix A.1 provides a formal unrolling demonstrating that the effective memory horizon expands as evidence accumulates. It further includes a rigorous proof of the $1/t$ gain decay in static environments and a proposition establishing the relationship between the steady-state gain and the noise ratio q/r.

4 Experiments

4.1 Setup

Model and update rules. We evaluate FILT3R as a drop-in replacement for the state update in a CUT3R-style recurrent architecture [33]. All methods share identical backbone weights; only the online update policy differs. FILT3R uses the token-wise filtering update in Eq. (5) with adaptive process noise Eq. (7) and gain computation in Eq. (4).

Tasks and metrics. We report (i) video depth estimation with Abs Rel, δ<1.25, and log RMSE, (ii) camera pose estimation with ATE, RPE-translation, and RPE-rotation, and (iii) 3D reconstruction with Accuracy, Completeness, and Normal Consistency. For depth, we report both metric-scale and per-sequence scale-aligned metrics following common practice. For long-horizon benchmarks, we additionally visualize error versus sequence length to characterize drift.

Datasets. Short-horizon depth is evaluated on Sintel [2], Bonn [17], and KITTI [6] following prior protocols. Long-horizon behavior is evaluated on extended TUM [22] and Bonn [17] sequences substantially longer than the training horizon. 3D reconstruction quality is measured on 7-Scenes [21] and NRGBD [1]. Full dataset and preprocessing details are provided in the appendix.

Baselines. We compare against CUT3R [33] (uniform overwrite) and TTT3R [4] (heuristic confidence scaling). We additionally report streaming reconstruction baseline with an explicit spatial memory architecture (Point3R). Additional comparisons with TTT3R+Reset are provided in the appendix (Tab. 7 and Tab. 8).

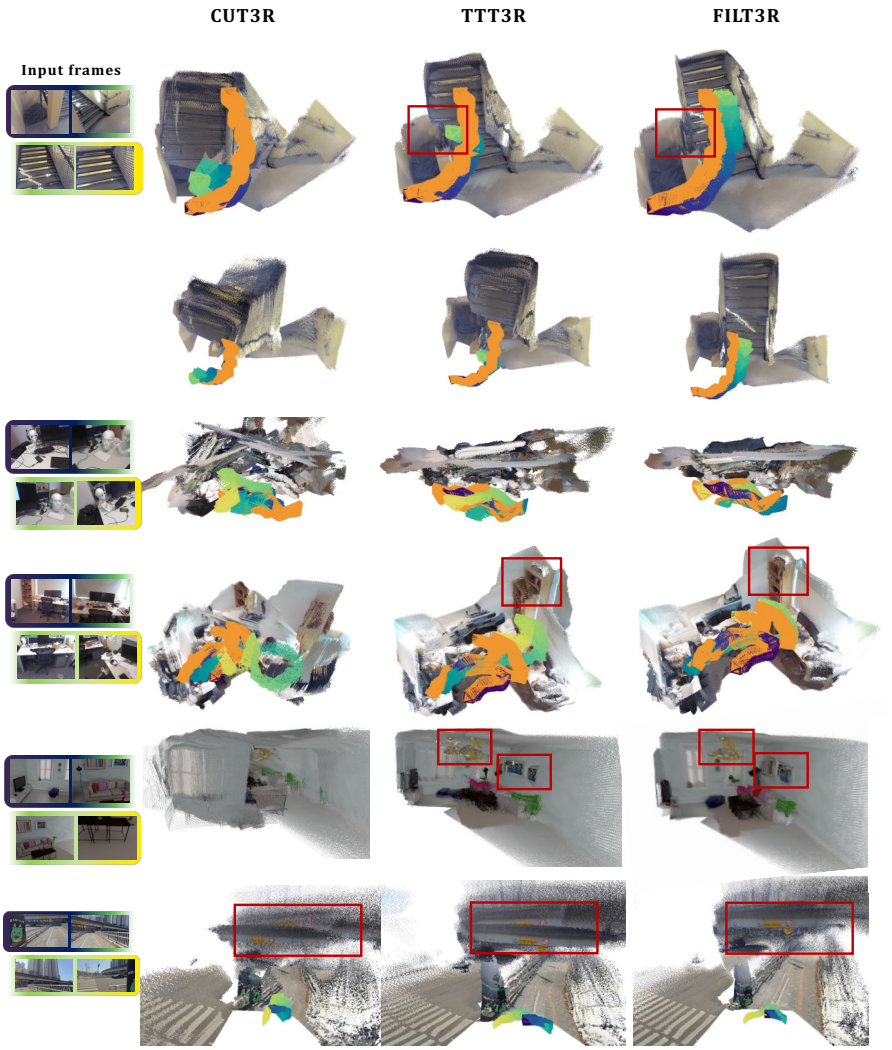


Fig. 3: Qualitative long-horizon streaming 3D reconstruction. CUT3R often suffers from catastrophic forgetting and drift, while TTT3R reduces but does not eliminate long-horizon instability, leading to fragmented surfaces and inconsistent revisited regions (red boxes). FILT3R produces more coherent geometry over long rollouts. Ground-truth trajectory is shown in orange; the predicted trajectory is color-coded by time.

Hyperparameter selection protocol. FILT3R has a small set of scalar hyperparameters controlling process and measurement noise. To avoid per-benchmark tuning, we tune on a held-out development set (long-horizon TUM-200 for pose,

Table 1: 3D reconstruction on 7-Scenes and NRGBD. Accuracy (Acc), Completeness (Comp), and Normal Consistency (NC); Mean and Median (Med.) are reported. Point3R runs out of memory (OOM) at length 1000.

(a) **7-Scenes** (lengths 300 / 500 / 1000)

Method	Length 300						Length 500						Length 1000					
	Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
CUT3R	0.134	0.090	0.076	0.041	0.544	0.565	0.183	0.133	0.093	0.036	0.530	0.544	0.233	0.170	0.101	0.017	0.511	0.515
Point3R	0.045	0.025	0.032	0.013	0.564	0.598	0.057	0.026	0.025	0.008	0.556	0.584	**OOM**					
TTT3R	0.040	0.025	0.024	0.005	0.566	0.602	0.066	0.038	0.032	0.006	0.551	0.576	0.145	0.092	0.051	0.010	0.525	0.536
FILT3R (ours)	**0.020**	**0.009**	**0.022**	**0.004**	**0.568**	**0.603**	**0.024**	**0.011**	**0.023**	**0.004**	**0.559**	**0.589**	**0.054**	**0.028**	**0.028**	**0.005**	**0.535**	**0.550**

(b) **NRGBD** (lengths 300 / 400 / 500)

Method	Length 300						Length 400						Length 500					
	Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑		Acc↓		Comp↓		NC↑	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
CUT3R	0.216	0.122	0.073	0.017	0.583	0.629	0.293	0.212	0.103	0.042	0.561	0.592	0.313	0.238	0.141	0.045	0.558	0.585
Point3R	**0.065**	0.037	**0.012**	**0.003**	0.617	0.693	**0.084**	0.040	**0.020**	**0.003**	0.613	0.684	0.112	0.046	**0.028**	**0.003**	0.614	0.687
TTT3R	0.102	0.044	0.024	0.004	0.613	0.684	0.144	0.064	0.072	0.011	0.598	0.656	0.170	0.095	0.090	0.019	0.590	0.642
FILT3R (ours)	0.079	**0.027**	**0.012**	0.004	**0.628**	**0.710**	0.086	**0.036**	0.026	0.005	**0.623**	**0.701**	**0.096**	**0.042**	0.035	0.005	**0.621**	**0.697**

and a short-horizon set for depth) and use the same hyperparameters across all the experiments.The list of hyperparameters is provided in the appendix

Origin-aligned ATE. Standard ATE applies a global alignment over the full trajectory, which can partially mask progressive drift. We therefore also report ATE_{orig}, aligning only the first frame and evaluating the remaining trajectory without further alignment, so drift accumulation is directly reflected in the error.

4.2 Long-horizon 3D reconstruction

Table 1 reports 3D reconstruction quality on 7-Scenes and NRGBD at long sequence lengths. On 7-Scenes, FILT3R consistently outperforms CUT3R and TTT3R, with the gap widening as sequences lengthen: mean accuracy improves from 0.040 to 0.020 at length 300 and from 0.066 to 0.024 at length 500 compared to TTT3R, while maintaining strong completeness and normal consistency. At length 1000—well beyond the training horizon—Point3R runs out of memory, while CUT3R and TTT3R degrade substantially; FILT3R continues to produce coherent geometry, confirming that uncertainty propagation compounds in value over long rollouts. On NRGBD, FILT3R improves over CUT3R and TTT3R at all lengths and remains competitive with prior streaming baselines at shorter horizons.

4.3 Long-horizon camera pose estimation

As shown in Table 2, FILT3R substantially reduces long-horizon trajectory error on extended TUM-RGBD sequences. The improvements are most pronounced under origin-aligned ATE, which directly exposes progressive drift: at 800 frames, ATE_{orig} drops from 0.214 (TTT3R) and 0.493 (CUT3R) to 0.107 with FILT3R.

Table 2: Long-horizon camera pose on TUM-RGBD. We report standard ATE (with global Procrustes alignment) and origin-aligned ATE (ATE_{orig}), which aligns only the first frame and therefore exposes progressive drift. FILT3R provides clear long-horizon stability gains.

Method	TUM-400				TUM-600				TUM-800			
	ATE↓	ATE_{orig}↓	RPE-t↓	RPE-r↓	ATE↓	ATE_{orig}↓	RPE-t↓	RPE-r↓	ATE↓	ATE_{orig}↓	RPE-t↓	RPE-r↓
CUT3R	0.109	0.302	0.010	0.381	0.145	0.425	**0.008**	0.430	0.173	0.493	**0.008**	0.486
TTT3R	0.055	0.128	0.010	0.328	0.084	0.176	0.009	0.365	0.097	0.214	0.009	0.387
FILT3R (ours)	**0.033**	**0.074**	**0.009**	**0.305**	**0.042**	**0.082**	0.009	**0.332**	**0.057**	**0.107**	0.009	**0.362**

Standard ATE also improves (0.097 → 0.057 at 800 frames). Local relative errors are similar to TTT3R in translation (RPE-t ≈ 0.009), while rotation error decreases (0.387 → 0.362 at 800 frames), suggesting FILT3R primarily mitigates slow drift and catastrophic forgetting rather than short-term relative motion.

4.4 Long-horizon video depth estimation

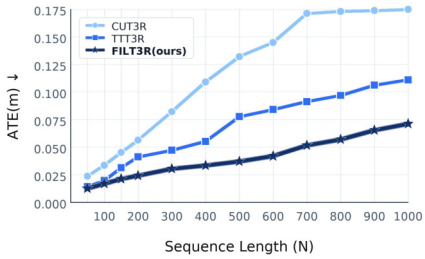
Table 3: Long-horizon video depth on Bonn (metric scale). Sequences are truncated to the indicated length. FILT3R maintains stable metric depth predictions as sequence length grows, whereas competing methods suffer from accumulating drift.

Method	Bonn-300			Bonn-400			Bonn-500		
	Abs Rel↓	δ<1.25 ↑	log RMSE↓	Abs Rel↓	δ<1.25 ↑	log RMSE↓	Abs Rel↓	δ<1.25 ↑	log RMSE↓
CUT3R	0.107	88.7	0.162	0.107	89.6	0.161	0.101	90.6	0.156
TTT3R	0.108	90.1	0.163	0.104	91.2	0.158	0.100	92.1	0.154
FILT3R (ours)	**0.093**	**93.8**	**0.152**	**0.090**	**94.1**	**0.149**	**0.089**	**94.4**	**0.148**

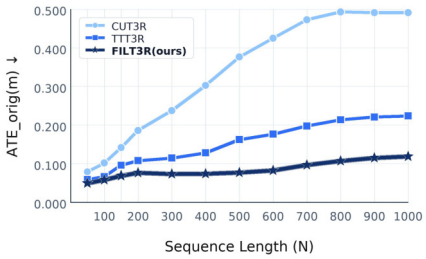
Table 3 evaluates metric-scale depth on extended Bonn sequences (300–500 frames), beyond the training horizon. FILT3R achieves the best performance at every truncation length. Compared to TTT3R, FILT3R reduces Abs Rel from 0.108 to 0.093 at 300 frames and from 0.100 to 0.089 at 500 frames, while improving δ<1.25 from 90.1 to 93.8 and from 92.1 to 94.4. The consistently lower log RMSE indicates improved robustness to rare but severe outlier predictions that can arise from overwriting a stable latent state with a noisy candidate update. Figure 4 (bottom) further shows FILT3R maintains lower error across sequence prefixes, indicating reduced drift accumulation.

4.5 Generality across memory architectures

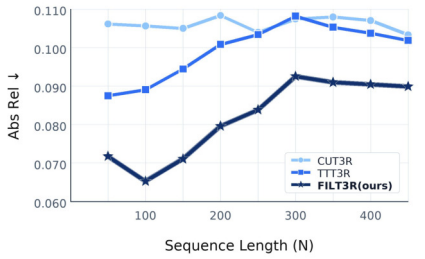
FILT3R is presented in a CUT3R-style setup, but the underlying idea—treating the recurrent state as a belief and the decoder candidate as a noisy measurement—is not specific to CUT3R's compact latent. We therefore apply it to two streaming models with *different* memory designs, changing only their state-update rule and



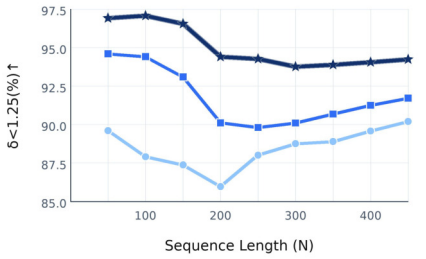
(a) TUM-RGBD: ATE vs. length



(b) TUM-RGBD: $\mathrm{ATE}_{\mathrm{orig}}$ vs. length



(c) Bonn: Abs Rel vs. length



(d) Bonn: δ<1.25 vs. length

Fig. 4: Long-horizon metrics vs. sequence length. Top (a, b): Camera pose on TUM-RGBD (50–1000 frames). FILT3R's drift grows more slowly. **Bottom (c, d):** Metric scale video depth on Bonn (50–500 frames). FILT3R consistently mitigates drift and improves over the baselines as the sequence progresses.

retraining nothing. Applied to the fast-weight memory of tttLRM [28] (inserted into its LaCT/Muon transition, denoted FILT3R-FW), FILT3R improves PSNR over the unmodified autoregressive update by up to +1.08 dB at length 128 and, when finer update steps induce catastrophic forgetting, rescues up to +1.6 dB on all 140 DL3DV-140 scenes (Fig. 5). It also extends to Point3R's [36] explicit pointer memory and compares favorably to the attention/KV-cache memory InfiniteVGGT [38], so FILT3R generalizes across *fast-weight*, *explicit-pointer*, and *compact-latent* memories. Full tables and analysis are in Appendix H (fast weights) and Appendix I (pointer memory).

4.6 Short-horizon regression check

Finally, we verify that FILT3R does not sacrifice short-horizon accuracy for long-horizon stability. Comparing the three update rules under identical backbone weights, FILT3R matches or improves short-horizon accuracy while providing clear long-horizon stability gains. Under metric-scale evaluation the gains are larger: Abs Rel drops from 1.029 (CUT3R) and 0.977 (TTT3R) to 0.772 on

Fig. 5: FILT3R on tttLRM fast-weight memory (DL3DV-140). Novel-view renders from GT, tttLRM, and tttLRM+FILT3R (top to bottom). Inserting FILT3R into the fast-weight update sharpens fine structure and suppresses the blur and ghosting that the unfiltered model accumulates over long view sequences; highlighted regions mark representative cases.

Sintel, and from 0.103 / 0.090 to 0.067 on Bonn. Full comparisons including optimization-based and full-attention baselines are in Appendix E.

4.7 Efficiency

Table 4 and Figure 6 compare runtime and memory. Full-attention architectures (VGGT, DA3) and Point3R scale linearly with sequence length, with Point3R running out of memory beyond ∼600 frames on a 32 GB GPU. FILT3R maintains effectively the same constant-memory footprint as CUT3R while TTT3R nearly doubles memory due to cached attention maps for attention-map-based gating.

4.8 Analysis

To understand why FILT3R improves long-horizon stability, we analyze its induced update dynamics on long TUM-RGBD sequences. FILT3R behaves as an uncertainty-calibrated conservative updater: as confidence accumulates, propagated uncertainty contracts and the effective gain settles into a low-update regime, which reduces the applied update magnitude. This matches the error profile in Table 2: FILT3R substantially lowers $\mathrm{ATE}_{\mathrm{orig}}$ while leaving short-horizon RPE nearly unchanged, indicating that its main benefit is not more aggressive correction at transition frames, but less persistent overwrite of the latent state and therefore slower cumulative drift. The adaptive process-noise term still matters because it temporarily relaxes this conservative regime on difficult frames, allowing the model to remain responsive without reverting to

Table 4: Runtime and GPU memory (500 frames). Benchmark on NRGBD (512×384, warmup=2, 10 runs). **FILT3R (ours)** avoids attention-map caching, matching CUT3R's footprint.

Method	Attn	FPS↑	Mem(MB)↓
CUT3R	–	25.27±.05	3503
TTT3R	✓	24.69±.05	6294
FILT3R (adpt. r_t)	✓	24.83±.04	6294
FILT3R (ours)	–	25.14±.04	3503

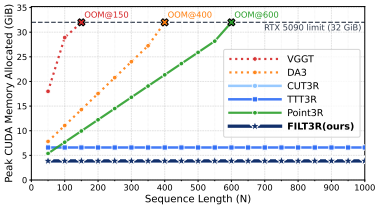


Fig. 6: Peak GPU mem. vs. frames. Full-attention methods (VGGT, DA3) and Point3R grow with length; CUT3R/FILT3R stay constant. FILT3R overlaps CUT3R, confirming negligible overhead.

Table 5: Ablation study of FILT3R components. All variants use identical frozen backbone weights and identical inference code; only the online update policy is changed. We report long-horizon pose on TUM-800 and long-horizon metric depth on Bonn-500. Lower is better (↓) except δ<1.25 (↑).

Variant	TUM-800 (pose)				Bonn-500 (metric depth)		
	ATE↓	ATE_{orig} ↓	RPE-t↓	RPE-r↓	AbsRel↓	δ<1.25 ↑	logRMSE↓
FILT3R (full)	0.057	0.107	0.009	0.362	0.089	94.4	0.148
Fixed-**Q** (fixed r)	0.100	0.229	0.009	0.393	0.099	91.9	0.155
No variance propagation (reset-**P**)	0.149	0.489	0.009	0.446	0.099	91.1	0.154
Fixed-β EMA (β=0.05, tuned)	0.049	0.078	0.010	0.658	0.093	93.5	0.152
No EMA normalization for drift	0.058	0.109	0.010	0.367	0.091	94.1	0.150
Adaptive measurement noise (r_t)	0.074	0.148	0.009	0.372	0.093	93.9	0.151

continual high-gain updates; consistent with this interpretation, removing uncertainty propagation in Table 5 substantially degrades long-horizon performance. Detailed transition-level statistics are provided in the appendix (Fig. 10).

4.9 Ablations

All variants in Table 5 share identical frozen backbone weights and identical inference code; we only change the *online state update*. FILT3R's update is fully determined by the gain $\boldsymbol{\beta}_t = \mathbf{k}_t$ in Eq. (5), which in turn depends on (i) the process noise $\mathbf{q}_t$ (Eq. (7)), (ii) the measurement noise r (Eq. (4)), and (iii) whether uncertainty is *propagated* through the variance recursion (Eqs. (3)–(6)). The ablations isolate each of these design choices.

(i) Adaptive process noise is necessary for targeted adaptation. **Fixed-Q** replaces the drift-driven $\mathbf{q}_t$ with a constant. This removes the ability to temporarily increase the update strength during genuine motion bursts or scene change. As a result, long-horizon drift grows substantially (*e.g.*, ATE_{orig} increases from 0.107 to 0.229), and depth accuracy also degrades (AbsRel 0.089 → 0.099). This

indicates that adaptivity should enter through the *process model* rather than a fixed gate.

(ii) Variance propagation enables confidence accumulation. **No variance propagation (reset-P)** keeps the same gain formula but discards the recursive uncertainty update. Without accumulated confidence, the gain cannot systematically shrink in stable intervals, so the model keeps injecting per-frame measurement noise into the state, leading to severe long-horizon drift (ATE_{orig} 0.107 $\rightarrow$ 0.489) and worse depth (AbsRel 0.089 $\rightarrow$ 0.099). This confirms that the Kalman-style *propagation* (not just the functional form of the gain) is important.

(iii) A tuned fixed-gain EMA can reduce ATE, but it harms local motion and depth. **Fixed-β EMA** uses a constant update $\mathbf{s}_t = (1-\beta)\mathbf{s}_{t-1} + \beta\,\tilde{\mathbf{s}}_t$ with $\beta = 0.05$ tuned on a development set. Interestingly, this can yield *lower* trajectory ATE on TUM-800 (0.049 vs. 0.057) and also improve ATE_{orig} (0.078 vs. 0.107), suggesting that a conservative fixed update can suppress some global drift. However, it does so by uniformly damping updates, which noticeably degrades *local* relative motion: rotation error increases sharply (RPE-r 0.362 $\rightarrow$ 0.658). At the same time, depth quality is still worse than FILT3R. Overall, fixed-β behaves like a *global low-pass filter*: it can reduce drift under some metrics, but it cannot simultaneously preserve accurate short-term motion and maintain depth quality across long rollouts. This highlights the need for *uncertainty-aware* and *transition-aware* gains rather than a single fixed forgetting rate.

(iv) EMA normalization is a stabilizer but not the dominant factor here. **No EMA normalization** removes the running baseline $\hat{\Delta}_t$ in Eq. (9). On these benchmarks, drift magnitudes are already relatively well-scaled, so the effect is modest (*e.g.*, ATE_{orig} 0.107 $\rightarrow$ 0.109; AbsRel 0.089 $\rightarrow$ 0.091). We keep EMA normalization as a robust default because it mitigates early/late scale shifts and reduces gain jitter when drift statistics vary across scenes and sequences.

(v) Adaptive measurement noise hurts long-horizon stability and adds overhead. **Adaptive** r_t replaces the fixed scalar r with a per-token measurement noise estimated from attention statistics, following the intuition that tokens attending strongly to image evidence should be trusted more (lower $r_{t,i}$). However, this variant degrades long-horizon performance compared to the full model: ATE_{orig} increases from 0.107 to 0.148 and ATE from 0.057 to 0.074, while depth also worsens (AbsRel 0.089 $\rightarrow$ 0.093, δ<1.25 94.4 $\rightarrow$ 93.9). We believe the reason is that attention-aligned novelty is correlated with temporal drift: during scene transitions, both $\mathbf{q}_t$ increases (from large drift) and $r_{t,i}$ decreases (from high attention to novel content), causing the gain to spike more aggressively than either signal alone would warrant. This co-excitation amplifies updates precisely when stability matters most, undermining the filter's conservative accumulation of confidence. Since adaptive r_t additionally requires caching attention maps—nearly doubling GPU memory to match attention-based gating baselines (Tab. 4)—we favor the simpler and more stable design: a fixed scalar r as a measurement-noise anchor, with all adaptivity concentrated in the process noise $\mathbf{q}_t$.

5 Limitations and discussion

FILT3R fixes r to a scalar and concentrates adaptivity in $\mathbf{q}_t$, which suits a frozen decoder but ignores genuine measurement-quality variation (blur, occlusion, motion into unseen regions); our ablation shows that estimating r_t from the same novelty signal that drives $\mathbf{q}_t$ *hurts* stability (Sec. 4.9), so reliable r_t adaptation likely needs an *external* cue, and because large drift only *raises* the gain FILT3R can over-trust the candidate when the backbone is least reliable. Our linear-Gaussian view is an inductive bias rather than an exact model of the nonlinear decoder—hence Kalman-*style*—and it assumes a streaming model exposing a candidate and a previous state; extending it to attention/KV-cache memories without an explicit recurrent state remains open.

6 Conclusion

We introduced FILT3R, a principled, training-free solution to the long-horizon drift and catastrophic forgetting that typically degrade streaming 3D reconstruction. By casting recurrent token updates as an adaptive Kalman-style filtering problem, FILT3R explicitly propagates per-token uncertainty to dynamically balance historical memory retention with fast adaptation. Crucially, this uncertainty-aware state fusion acts as a plug-and-play stabilizer. Relying solely on internal temporal drift to estimate process noise—anchored by a single, fixed measurement noise scalar—it unifies and outperforms prior heuristic update rules without requiring any retraining. Our extensive evaluations across video depth, camera pose, and 3D reconstruction demonstrate that FILT3R significantly extends the effective memory horizon, paving the way for highly stable, continuous, and long-form online scene understanding.

Acknowledgements

This work was supported by the National Research Foundation of Korea under Grant RS-2024-00336454, and by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST)). This research was also supported by the Advanced GPU Utilization Support Program funded by the Government of the Republic of Korea (Ministry of Science and ICT) (02-26-01-0404), and by the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT), Republic of Korea (RQT-25-120217).

References

1. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural rgb-d surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6290–6301 (June 2022)
2. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 611–625 (2012)
3. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Easi3R: Estimating disentangled motion from DUSt3R without training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9158–9168 (2025), https://openaccess.thecvf.com/content/ICCV2025/html/Chen_Easi3R_Estimating_Disentangled_Motion_from_DUSt3R_Without_Training_ICCV_2025_paper.html
4. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: TTT3R: 3d reconstruction as test-time training. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=aMs6FtNaY5
5. Deng, K., Ti, Z., Xu, J., Yang, J., Xie, J.: Vggt-long: Chunk it, loop it, align it – pushing vggt's limits on kilometer-scale long rgb sequences (2025), https://arxiv.org/abs/2507.16443
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3354–3361 (2012). https://doi.org/10.1109/CVPR.2012.6248074, https://www.cvlibs.net/publications/Geiger2012CVPR.pdf
7. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Backprop KF: Learning discriminative deterministic state estimators. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 29 (2016), https://proceedings.neurips.cc/paper/2016/hash/697e382cfd25b07a3e62275d3ee132b3-Abstract.html
8. Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering **82**(1), 35–45 (1960)
9. Krishnan, R.G., Shalit, U., Sontag, D.: Deep kalman filters (2015), https://arxiv.org/abs/1511.05121
10. Lan, Y., Luo, Y., Hong, F., Zhou, S., Chen, H., Lyu, Z., Dai, B., Yang, S., Loy, C.C., Pan, X.: STream3R: Scalable sequential 3d reconstruction with causal transformer. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=RTTYGeC2Io, poster
11. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 71–91 (2024). https://doi.org/10.1007/978-3-031-73220-1_5, https://link.springer.com/chapter/10.1007/978-3-031-73220-1_5
12. Li, Z., Zhou, J., Wang, Y., Guo, H., Chang, W., Zhou, Y., Zhu, H., Chen, J., Shen, C., He, T.: Wint3r: Window-based streaming reconstruction with camera token pool. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=PjviszIZf1
13. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Zhao, Y., Shi, G., Feng, J., Kang, B., Peng, S., Guo, H., Zhou, X.: Depth anything 3: Recovering the visual space from any views. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=yirunib8l8
14. Liu, Y., Dong, S., Wang, S., Yin, Y., Yang, Y., Fan, Q., Chen, B.: Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16651–16662 (2025)
15. Mehra, R.K.: On the identification of variances and adaptive kalman filtering. IEEE Transactions on Automatic Control **15**(2), 175–184 (1970). https://doi.org/10.1109/TAC.1970.1099422
16. Murai, R., Dexheimer, E., Davison, A.J.: Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16695–16705 (2025)
17. Palazzolo, E., Behley, J., Lottes, P., Giguère, P., Stachniss, C.: ReFusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7855–7862 (2019). https://doi.org/10.1109/IROS40897.2019.8967590, https://www.ipb.uni-bonn.de/pdfs/palazzolo2019iros.pdf
18. Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: Point-slam: Dense neural point cloud-based SLAM. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 18433–18444 (2023)
19. Shen, G., Deng, T., Qin, X., Wang, N., Wang, J., Wang, Y., Chen, Y., Wang, H., Wang, J.: Mut3r: Motion-aware updating transformer for dynamic 3d reconstruction (2025), https://arxiv.org/abs/2512.03939
20. Shen, Y., Zhang, Z., Qu, Y., Zheng, X., Ji, J., Zhang, S., Cao, L.: Fastvggt: Fast visual geometry transformer. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=asl8NJlIMe
21. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2930–2937 (2013)
22. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 573–580 (2012)
23. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: iMAP: Implicit mapping and positioning in real-time. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6229–6238 (2021)
24. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A.A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 9229–9248 (2020), https://proceedings.mlr.press/v119/sun20b.html
25. Teed, Z., Deng, J.: Deepv2d: Video to depth with differentiable structure from motion. In: International Conference on Learning Representations (ICLR) (2020), https://openreview.net/forum?id=HJeO7RNKPr
26. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
27. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 16558–16569. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/89fcd07f20b6785b92134bd6c1d0fa42-Paper.pdf
28. Wang, C., Tan, H., Yifan, W., Chen, Z., Liu, Y., Sunkavalli, K., Bi, S., Liu, L., Hu, Y.: tttlrm: Test-time training for long context and autoregressive 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 36582–36592 (June 2026)

29. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: International Conference on Learning Representations (ICLR) (2021), https://arxiv.org/abs/2006.10726
30. Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: 2025 International Conference on 3D Vision (3DV). pp. 78–89 (2025). https://doi.org/10.1109/3DV66043.2025.00013
31. Wang, H., Wang, J., Agapito, L.: Co-SLAM: Joint coordinate and sparse parametric encodings for neural real-time SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13293–13302 (2023)
32. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5294–5306 (2025)
33. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10510–10522 (2025)
34. 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 (CVPR). pp. 20697–20709 (2024)
35. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=DTQIjngDta, poster
36. Wu, Y., Zheng, W., Zhou, J., Lu, J.: Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. In: Advances in Neural Information Processing Systems (NeurIPS) (2025), https://openreview.net/forum?id=yk1iqV9Etr
37. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21924–21935 (2025)
38. Yuan, S., Yang, Y., Yang, X., Zhang, X., Zhao, Z., Zhang, L., Zhang, Z.: Infinitevggt: Visual geometry grounded transformer for endless streams (2026), https://arxiv.org/abs/2601.02281
39. 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: International Conference on Learning Representations (ICLR) (2025), https://openreview.net/forum?id=lJpqxFgWCM, spotlight
40. Zheng, Z., Xiang, X., Zhang, J.: TTSA3R: Training-free temporal-spatial adaptive persistent state for streaming 3d reconstruction (2026), https://arxiv.org/abs/2601.22615
41. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: NICE-SLAM: Neural implicit scalable encoding for SLAM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12786–12796 (2022)
42. Zhuo, D., Zheng, W., Guo, J., Wu, Y., Zhou, J., Lu, J.: Streaming visual geometry transformer. In: International Conference on Learning Representations (ICLR) (2026), https://openreview.net/forum?id=5APgTKsnx8