VMem: Consistent Interactive Video Scene Generation with Surfel-Indexed View Memory

Runjia Li Philip Torr Andrea Vedaldi Tomas Jakab
University of Oxford
v-mem.github.io

Appendix

A. Implementation details

To address the computational cost of SEVA [2], which uses a large fixed total number of reference and target frames ($K + M = 21$), we fine-tune a more efficient version that employs a reduced number of reference frames ($K = 4$) and target views ($M = 4$). We apply LoRA [1] with rank 256 and randomly sample context views online during training. Training proceeds for 600,000 iterations on 8 A40 GPUs with a batch size of 24 per GPU, using the AdamW optimizer with a learning rate of 3×10^{-6}, weight decay of 10^{-4}, and a cosine annealing schedule. For inference, we set the classifier-free guidance scale to 3, the point map scaling factor σ to 0.03, and α to 0.2 for surfel radius calculation.

B. Average pose calculation

To compute the average camera pose for rendering surfels, we average translations $\mathbf{t}_{T+m}$ with a simple mean, and rotations $\mathbf{R}_{T+m}$ by converting them to quaternions $\mathbf{q}_m$, aligning signs to a common hemisphere, and normalizing the mean quaternion:

$$\bar{\mathbf{q}} = \frac{\sum_{m=1}^{M} \tilde{\mathbf{q}}_m}{\| \sum_{m=1}^{M} \tilde{\mathbf{q}}_m \|}, \quad \tilde{\mathbf{q}}_m = \text{sign}(\mathbf{q}_m \cdot \mathbf{q}_1) \cdot \mathbf{q}_m.$$

The final average pose is $\bar{\mathbf{c}} = \begin{bmatrix} \mathrm{R}(\bar{\mathbf{q}}) & \bar{\mathbf{t}} \\ \mathbf{0}^\top & 1 \end{bmatrix}$, where $\mathrm{R}(\bar{\mathbf{q}})$ denotes the rotation matrix from $\bar{\mathbf{q}}$ and $\bar{\mathbf{t}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{t}_{T+m}$.

C. Autoregressive point map prediction

Since we generate point maps for each view in an autoregressive manner, it is crucial to maintain their consistency across a shared coordinate space. Point-map estimators such as CUT3R include an optimization stage that jointly refines the depth, camera parameters, and point maps. To ensure a fixed camera trajectory, we freeze the camera parameters, which are user-defined inputs. Additionally, at each generation step when we have T frames generated so far, we freeze all previously predicted depth maps for frames $1, 2, \ldots, T$ during optimization. This ensures that the resulting point maps and surfel representations remain consistent and causal. We then save the optimized depth maps of the newly generated frames $T + 1, \ldots, T + M$ for future prediction.

D. Limitations and discussion

Evaluation protocol. Since there is no established benchmark for evaluating long-term consistency in scene video generation, we adopt cyclic trajectories as a proxy for assessment. However, these trajectories remain relatively simple and contain only limited occlusions, which means the full potential of VMem in handling occlusions is not fully demonstrated. Moreover, existing evaluation metrics primarily capture low-level texture similarity in hallucinated content, rather than assessing true multi-view consistency—an inherent limitation of single-view autoregressive generation. As such, there is a clear need for more standardized evaluation protocols, which we leave for future exploration.

Limited training data and computing resources. Due to limited computational resources, our more efficient version of the generator based on SEVA [2] was fine-tuned only on the RealEstate10K dataset [3]. This dataset primarily consists of indoor scenes and a limited number of outdoor real-estate scenarios. Consequently, the model may struggle to generalize to broader contexts, with performance potentially degrading when dealing with natural landscapes or images containing moving objects compared to indoor environments. We believe this limitation stems primarily from insufficient dataset diversity rather than fundamental model constraints.

Inference speed. Due to the multi-step sampling process of diffusion models, VMem requires 4.16 seconds to generate a single frame on an RTX 4090 GPU. This falls short of the real-time performance needed for applications such as virtual reality. We believe that future advancements in single-step image-set models and improvements in computational infrastructure hold promise for significantly accelerating inference speed.

Future improvements. Since our memory module relies heavily on the capabilities of the off-the-shelf image-set generator and the point map predictor, the performance of VMem is expected to improve as these underlying models continue to advance.

References

[1] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*, 2022. 1

[2] Jensen (Jinghao) Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. 1

[3] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 1