This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

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

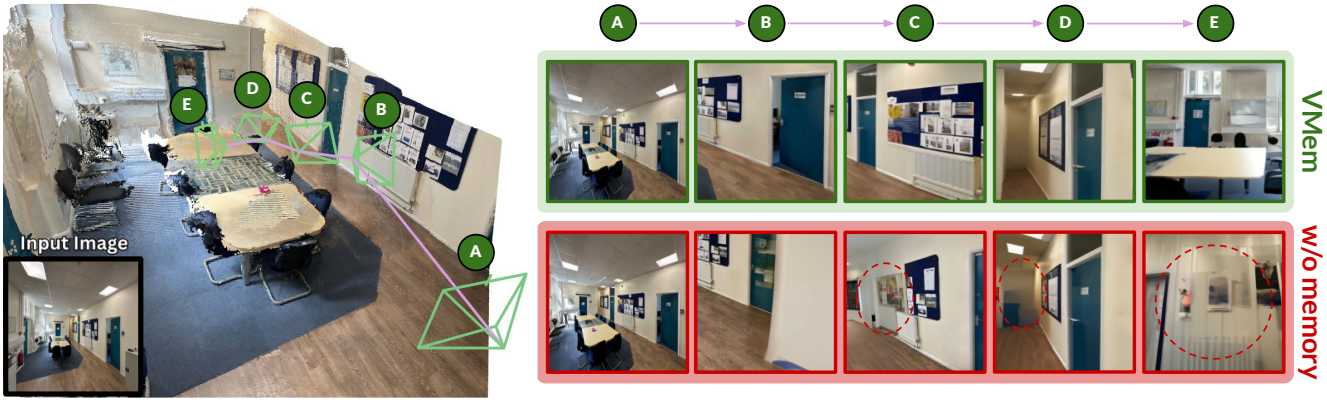


Figure 1. VMem enables autoregressive scene generation from a single image along user-defined trajectories. The green region shows results with the proposed memory module, maintaining coherence when generating previously seen parts of the scene. The red region, without memory, exhibits degradation highlighted with red ellipses, demonstrating VMem is effective for consistent scene generation.

Abstract

We propose a novel memory module for building video generators capable of interactively exploring environments. Previous approaches have achieved similar results either by out-painting 2D views of a scene while incrementally reconstructing its 3D geometry—which quickly accumulates errors—or by using video generators with a short context window, which struggle to maintain scene coherence over the long term. To address these limitations, we introduce Surfel-Indexed View Memory (VMem), a memory module that remembers past views by indexing them geometrically based on the 3D surface elements (surfels) they have observed. VMem enables efficient retrieval of the most relevant past views when generating new ones. By focusing only on these relevant views, our method produces consistent explorations of imagined environments at a fraction of the computational cost required to use all past views as context. We evaluate our approach on challenging long-term scene synthesis benchmarks and demonstrate superior performance compared to existing methods in maintaining scene coherence and camera control.

1. Introduction

We consider the problem of generating long videos that explore an imagined space following a camera path specified interactively by the user. In this paradigm, the user tells the model which camera path to follow for the next few frames, observes the generated content, and then decides where to explore next based on what they have seen. For example, exploring a house may involve visiting the kitchen, the living room, and the bathroom, eventually returning to the kitchen. Throughout the video, the scene must remain consistent, ensuring that the kitchen looks the same upon return. Generating such videos is essential for immersive applications such as games, where players can navigate generated worlds. However, even recent large-scale interactive video generators such as Google's Genie 2 [20] struggle to achieve this goal.

This problem has so far been addressed by two types of methods. First, *outpainting-based* methods [7, 11, 15, 16, 18, 21, 24, 26, 36, 43] iterate between generating new 2D views of the scene and estimating its 3D geometry. They use the estimated geometry to partially render a new viewpoint and then employ an outpainting model to complete the missing parts, thus adding one more image to the col-

lection. However, errors in outpainting, 3D reconstruction, and stitching accumulate over time, leading to severe degradation of the generated content after a short while.

Second, *multi-view/video-based* methods [20, 22, 25, 35, 42] condition novel view generation on previous views using a geometry-free approach that does not explicitly estimate the scene geometry. While this avoids the accumulation of errors in the reconstructed 3D scene geometry, it comes at a high computational cost, limiting the number of conditioning views to a small context window of recent frames. This constraint hurts the long-term consistency of the generated images.

In this work, we revisit the second class of methods and propose conditioning not on the most recently generated views, but on the *most relevant ones* for generating the next image, thereby maintaining a high degree of consistency within a limited computational budget. Given a novel viewpoint of the scene, the most relevant past views are those that have already observed the parts of the scene currently being generated. This implies that, for each part of the scene, we must remember which views have previously observed it and retrieve them from memory.

To achieve this, we introduce *Surfel-Indexed View Memory*, abbreviated as *VMem*, a memory module that anchors previous views to the 3D surface elements they observe. Given a new viewpoint, we retrieve the past views that best capture the currently observed surfaces and use them to condition novel view generation. To create the memory, we estimate the geometry of each new view using an off-the-shelf point map predictor. This is similar to outpainting methods but, crucially, we do not use the estimated geometry as the final representation of the scene; instead, we use it to construct VMem, which is a memory of past views. We represent the scene geometry using surfels, which are more robust compared to meshes and can represent occlusions compared to point clouds. Each surfel stores in its attributes a set of indices corresponding to past viewpoints that observed it.

To retrieve relevant past viewpoints, we render the surfels with their attributes from the novel viewpoint and splat them onto an image grid. Each pixel in the resulting image then corresponds to a set of viewpoint indices. We select the top K most frequently represented viewpoint indices and use them to retrieve past views from the database, where each view is represented by an RGB image and its corresponding camera parameters. A key advantage of our approach over outpainting-based methods is that it does not require highly accurate scene geometry. As long as we successfully retrieve the most relevant past views, our method remains robust.

By leveraging *Surfel-Indexed View Memory*, we significantly reduce the memory and computational burden of conditioning on a large number of previous views, as required by multi-view/video-based methods, while improving long-term consistency across generated novel views.

Our approach represents a step toward scalable, realistic, and long-term autoregressive scene generation, making the following contributions:

1. We introduce Surfel-Indexed View Memory, a *plug-and-play* module to index past views geometrically and use them to condition novel view generation.
2. We show that our method can generate long-term coherent videos of scenes and outperforms existing approaches.
3. We demonstrate that Surfel-Indexed View Memory achieves comparable performance with $4\times$ fewer context views, delivering a $12\times$ speedup.
4. We validate our approach on challenging benchmarks, outperforming the current open-source state-of-the-art methods.

2. Related work

Novel view synthesis (NVS) is a challenging and ill-posed problem that can be categorized into two main categories: view interpolation methods [3–6, 8, 19, 30, 37], which generate views close to given inputs, and view extrapolation or autoregressive view generation, where novel views extend significantly beyond the original scene, introducing substantial new content. The latter is particularly difficult from single images. Between these lie single-view scene reconstruction methods, which succeed mainly for single-object scenes [1, 12, 17, 27, 29, 38] or highly bounded scenes [28]. However, their extrapolation capabilities remain limited. Most relevant to our work are single-image view extrapolation models, which fall into two categories: those incorporating explicit geometric modeling with inpainting, and those based directly on image or video generation.

Inpainting-based view extrapolation. Inpainting-based methods [7, 11, 15, 16, 18, 21, 24, 26, 36, 40, 41, 43] use pre-trained 3D reconstruction models to generate 3D representations—such as meshes, point clouds, or Gaussian splats—from images. These representations are reprojected into novel views, where 2D inpainting fills missing regions. SceneScape [7] reconstructs meshes from images using pre-trained depth estimators. Diffusion-based inpainting then completes the projected novel view, which is reprojected back to refine and extend the mesh. This process is iterated to generate novel views. MultiDiff [18] uses depth estimators to warp reference images into multiple novel views, training diffusion models to inpaint missing regions simultaneously. CamCtrl3D [21] follows a similar approach but adds ray maps to condition diffusion models for view synthesis. ViewCrafter [43] uses pre-trained pointmap estimators to create point clouds and trains video diffusion mod-

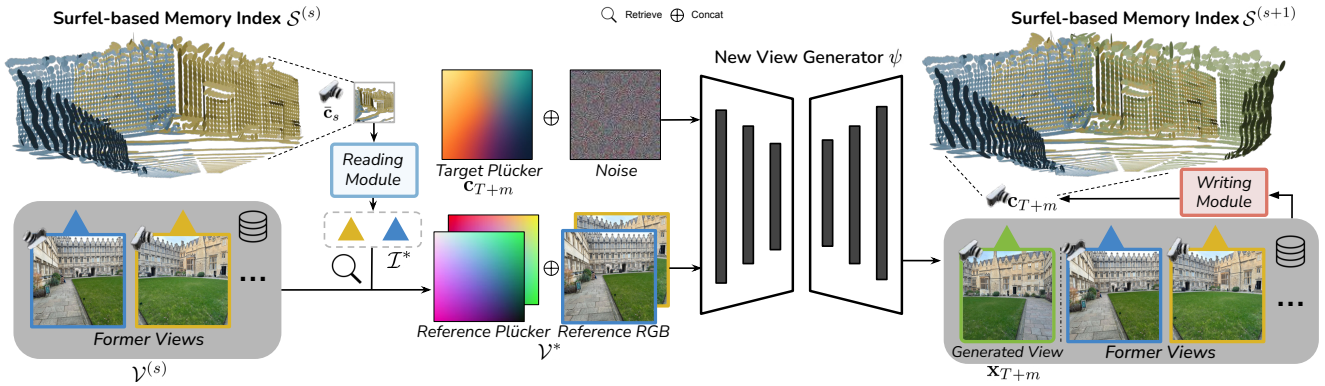


Figure 2. **Method.** Given target camera viewpoints $\{\mathbf{c}_{T+m}\}_{m=1}^{M}$, we query our Surfel-Indexed View Memory to retrieve the most relevant K past views $\mathcal{V}^* \subset \mathcal{V}^{(s)}$ where $\mathcal{V}^* = \{v_t\}_{t=1}^{K}$ as references. Retrieved reference images $\mathbf{x}_t$ along with Plücker embeddings of both reference camera poses $\mathbf{c}_t$ and target camera poses $\{\mathbf{c}_{T+m}\}_{m=1}^{M}$ are fed into generator ψ to synthesize novel views $\{\mathbf{x}_{T+m}\}_{m=1}^{M}$. After generation, the surfel-indexed memory is updated $\mathcal{S}^{(s)} \to \mathcal{S}^{(s+1)}$ by appending new view indices $\{T+m\}_{m=1}^{M}$ to existing surfels or creating new surfels based on geometry of the generated views. This is repeated autoregressively, enabling long-term consistent generation.

els for inpainting. GenWarp [26] warps 2D coordinate representations of reference views into novel views and uses them to condition generation.

While these methods improve geometric consistency, they are susceptible to errors in depth or pointmap estimation, which propagate distortions across generated views. Once 3D representations are constructed, inaccuracies from depth or pointmap estimation become difficult to correct. Moreover, scaling to large scenes is computationally intensive, as storing and processing high-fidelity 3D representations requires substantial memory.

Multi-view-based view extrapolation. Another category [8, 22, 25, 31, 35, 42, 46] avoids explicitly modeling 3D scene geometry. Instead, these approaches generate novel views by conditioning on previously rendered or generated views. GeoGPT [25] uses autoregressive likelihood models to synthesize novel views from single images. Building on this, LookOut [22] improves consistency by generating view sequences along predefined trajectories, conditioning on up to two preceding views. Other methods [31, 42] enhance novel view synthesis by employing cross-view attention or enforcing epipolar constraints within diffusion models. MotionCtrl [35] leverages geometric priors from pre-trained video diffusion models to synthesize camera-controllable videos. These methods are *trajectory-based*, requiring input views to follow continuous spatial paths. In contrast, *image-set models* like CAT3D [8] and more recently SEVA [46] synthesize novel views from sparse and unordered input images, without trajectory constraints. However, these methods are computationally expensive, with $O(n^2)$ complexity for attention, limiting conditioning views to small context windows containing only recently generated frames. This limitation impacts long-term consistency of generated images.

Several concurrent works explore memory-based approaches for scene generation. Genie [2, 20] maintains memory through recurrent state features. Gen3C [23] conditions on stored point clouds like ViewCrafter [43], inheriting inpainting-based limitations. StarGen [44] and WorldMem [39] use distance- and field-of-view-based spatial memory retrieval, which can recover spatially correlated views but lack geometric reasoning for occlusions. In contrast, our surfel-based memory explicitly models occlusions and provides more principled view selection.

3. Method

Let $\mathbf{x}_1 \in \mathbb{R}^{H\times W\times 3}$ be an RGB image of a scene that we wish to explore, and let $\{\mathbf{c}_t\}_{t=1}^{T}$ be a sequence of camera parameters specifying a path through the scene. Our goal is to generate a corresponding video, *i.e.*, a sequence of images $\{\mathbf{x}_t\}_{t=1}^{T}$, where $\mathbf{x}_1$ is the given input image, and each subsequent frame $\mathbf{x}_t$ is consistent with the previous ones and reflects the specified cameras $\{\mathbf{c}_t\}_{t=1}^{T}$. Moreover, we aim to do so autoregressively, generating M novel views at a time for the given cameras (with M kept small to ensure interactivity) while ensuring consistency with previously generated views. This enables interactive scene exploration, where the user can specify the next camera positions at each step.

Specifically, we work with a sequence of *views*, where each view $v_t = (\mathbf{x}_t, \mathbf{c}_t)$ consists of an image $\mathbf{x}_t$ and its corresponding camera $\mathbf{c}_t$ at timestep t. At generation step s ($s \geq 0$), we have generated $T = sM$ frames so far. The goal is to generate M new RGB images $\{\mathbf{x}_{T+m}\}_{m=1}^{M}$ for the next target camera parameters $\{\mathbf{c}_{T+m}\}_{m=1}^{M}$, given the previously generated views $\mathcal{V}^{(s)} = \{v_1, v_2, \ldots, v_T\}$. After generation, we update $T \leftarrow T + M$.

A challenge with this approach is that the context $\mathcal{V}^{(s)}$

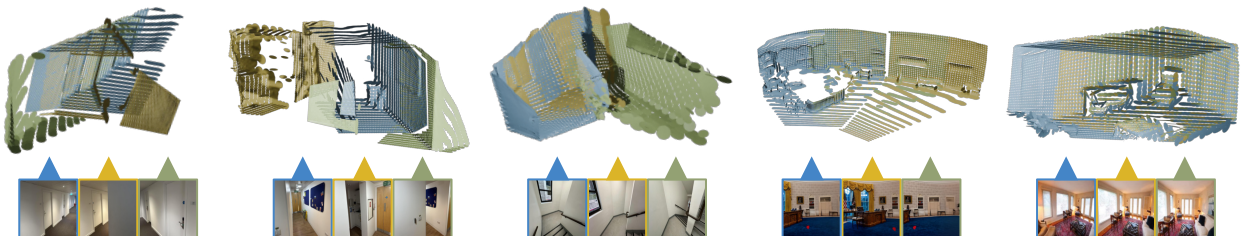


Figure 3. **Surfel-based memory index.** Each surfel stores indices of views that observed it. We color-code each surfel by contributing view indices. This spatial index enables retrieval of relevant past views: when generating a novel view, we identify visible surfels from the target viewpoint and retrieve views that previously observed those same regions, naturally accounting for occlusion.

grows unbounded over time. Video generators addressing this problem [22, 25, 35, 42] typically consider a fixed-length subset of $\mathcal{V}^{(s)}$, conditioning their generation only on the most recent L views $\{v_{T-L+1}, \ldots, v_T\}$. This limitation results in severe inconsistencies when the generated sequence extends beyond the (small) context window.

We solve this problem by dynamically retrieving the most relevant subset of past views, denoted as $\mathcal{V}^* \subseteq \mathcal{V}^{(s)}$. To achieve this, we introduce an efficient data structure, the Surfel-Indexed View Memory, that stores and retrieves past views based on their approximate 3D geometry. We first describe the Surfel-Indexed View Memory (Sec. 3.1) and then introduce the novel view generator (Sec. 3.2). An overview of the method is provided in Fig. 2.

3.1. Surfel-indexed view memory

Consider the problem of generating the next views $\{\mathbf{x}_{T+m}\}_{m=1}^{M}$ of the scene as seen from the cameras $\{\mathbf{c}_{T+m}\}_{m=1}^{M}$. The generator must consider the previously generated views and ensure $\{\mathbf{x}_{T+m}\}_{m=1}^{M}$ are consistent with those, while generating new content as needed. However, not all past views are equally relevant to the novel views $\{\mathbf{x}_{T+m}\}_{m=1}^{M}$. We prioritize past views that have likely observed the largest portion of the scene currently being generated. For example, if the current locale is surrounded by walls, parts of the scene behind those walls are likely less relevant for generating a new view of that locale. Conversely, views that share a similar visible region with the target view can provide more useful information for generating its content.

This principle drives our design for a view memory module. We maintain a coarse model of the scene geometry using surfels (visualized in Fig. 3), simple surface primitives that account for occlusion. Each surfel stores the indices of the past views that observed it. To generate a new view, the surfels visible from the new viewpoint are retrieved, using the associated indices to vote for which views to consider for generation. This process is illustrated in Fig. 4.

Specifically, we define a *surfel* as the tuple

$$\mathbf{s}_k = (\mathbf{p}_k,\, \mathbf{n}_k,\, r_k,\, \mathcal{I}_k)\,,$$

where $\mathbf{p}_k \in \mathbb{R}^3$ denotes the surfel's 3D position, $\mathbf{n}_k \in \mathbb{R}^3$ is its surface normal, $r_k \in \mathbb{R}$ is the surfel radius, and $\mathcal{I}_k \subseteq \{1, 2, \ldots, T\}$ is a subset of past view indices that observed the surfel. The surfel-based memory indexing of past views $\mathcal{V}^{(s)}$, encapsulating the scene, is represented as a set of surfels $\mathcal{S}^{(s)} = \{\mathbf{s}_k\}_{k=1}^{N^{(s)}}$, where $N^{(s)}$ is the number of surfels at generation step s. We also maintain an octree to quickly retrieve surfels based on their geometry. Next, we describe how to read from and write to this memory.

Reading from the memory. To retrieve the most relevant past views from the memory $\mathcal{V}^{(s)}$ for the novel cameras $\{\mathbf{c}_{T+m}\}_{m=1}^{M}$, we first compute their average pose $\bar{\mathbf{c}}_s \in \mathrm{SE}(3)$ as a reference (refer to the supplementary material). Then we render the surfels $\mathcal{S}^{(s)}$ from the averaged camera pose $\bar{\mathbf{c}}_s$. Each surfel is rendered as a splat with its attributes, accounting for relative depth and occlusions, and contributing to the image based on its coverage. The rendered attributes correspond to the indices of the past views. The core intuition is that views observing the largest portion of the scene from the perspective of the novel camera are most relevant for novel view synthesis. To identify these views, we rank past view indices based on their frequency across all rendered pixels.

We then select the top-K most frequently represented indices $\mathcal{I}^*$ and use them to retrieve the corresponding past views, $\mathcal{V}^* = \{v_t\}_{t \in \mathcal{I}^*}$, from the memory $\mathcal{V}^{(s)}$. This subset of views serves as the context for the new view generator, as described in Sec. 3.2.

To avoid oversampling repeatedly visited regions, we apply a non-maximum suppression algorithm that reduces redundancy in memory and promotes broader scene coverage among the top-K views. During retrieval, we retain only the most frequently referenced view among those with similar poses. During writing to the memory, we merge surfels by comparing their associated camera poses—if two poses are highly similar, we discard the older one.

Writing to the memory. After generating the new views $\{v_{T+m} = (\mathbf{x}_{T+m}, \mathbf{c}_{T+m})\}_{m=1}^{M}$, we add them to the memory $\mathcal{V}^{(s)}$ and update the surfel-based memory index $\mathcal{S}^{(s)}$. We use an off-the-shelf point map estimator ϕ, such as

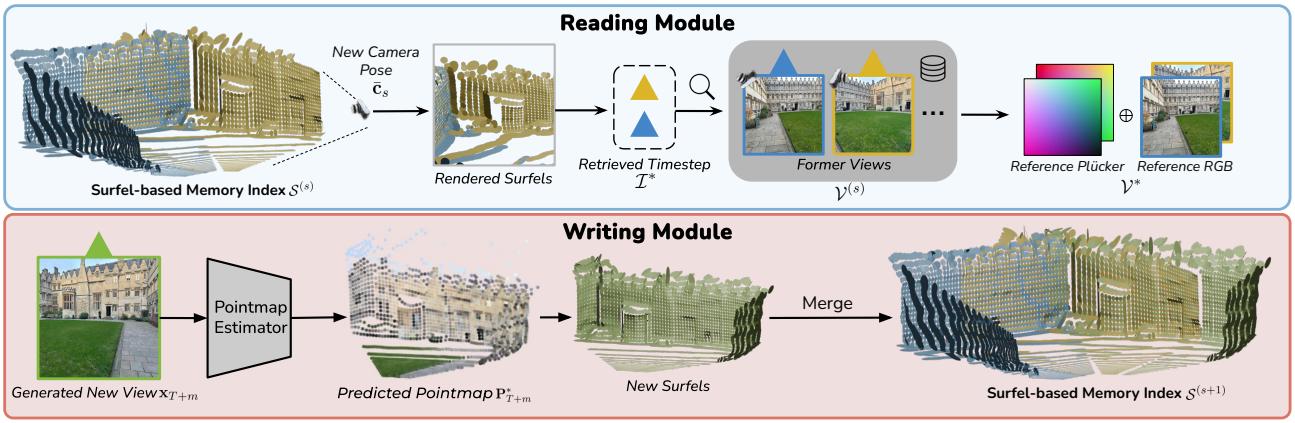


Figure 4. **Surfel-Indexed View Memory.** *Reading* procedure renders surfels $\mathcal{S}^{(s)}$ with their attributes, containing past view indices as frame indices. We then select the K most frequent frame indices in the rendered image to retrieve relevant past views from $\mathcal{V}^{(s)}$. *Writing* procedure estimates geometry of newly generated views $\{\mathbf{x}_{T+m}\}_{m=1}^{M}$ as surfels and merges them with existing surfels. Frame indices $\{T+m\}_{m=1}^{M}$ are appended to surfels in these views, and novel views are stored, updating $\mathcal{V}^{(s)} \to \mathcal{V}^{(s+1)}$ and $\mathcal{S}^{(s)} \to \mathcal{S}^{(s+1)}$.

CUT3R [32], to estimate their point maps $\{\mathbf{P}^*_{T+m}\}_{m=1}^{M}$. Specifically, we jointly estimate the point maps for the newly generated views along with the retrieved past views $\mathcal{V}^*$. This joint estimation ensures that the new point maps are aligned with the coordinate frame of the existing scene geometry in the surfel-based memory index $\mathcal{S}^{(s)}$.

Next, we convert the estimated point maps to a set of new surfels. For each newly generated frame $t \in \{T+1, T+2, \ldots, T+M\}$, we process its point map $\mathbf{P}^*_t$. Since our approach only requires coarse geometry, we first downsample the point map $\mathbf{P}^*_t$ by a factor σ, yielding a smaller point map $\mathbf{P}_t \in \mathbb{R}^{H' \times W' \times 3}$, where $H' = H/\sigma$ and $W' = W/\sigma$. For each pixel location (u, v) in the downsampled point map, we have a 3D point $\mathbf{p}_{u,v,t}$, and we create a new surfel $\mathbf{s}_{k'}$ centered on it. We compute each surfel's normal using the cross-product of displacement vectors to neighboring pixels. For a pixel at 2D location (u, v) in the image grid, we use neighboring pixels to estimate the local surface normal:

$$\mathbf{n}_{k'} = \frac{(\mathbf{p}_{u+1,v,t} - \mathbf{p}_{u-1,v,t}) \times (\mathbf{p}_{u,v+1,t} - \mathbf{p}_{u,v-1,t})}{\|(\mathbf{p}_{u+1,v,t} - \mathbf{p}_{u-1,v,t}) \times (\mathbf{p}_{u,v+1,t} - \mathbf{p}_{u,v-1,t})\|},$$

where $\mathbf{p}_{u,v,t}$ represents the 3D point at image coordinates (u, v) in frame t. To ensure that the surfels reasonably cover the scene, we use a heuristic to compute the surfel's radius, which is proportional to the depth of the surfel and inversely proportional to the focal length and the cosine of the angle between the surfel normal and the viewing direction:

$$r_{k'} = \frac{\frac{1}{2}\mathbf{D}_{u,v,t}/f_t}{\alpha + (1-\alpha)\,\|\mathbf{n}_{k'} \cdot (\mathbf{p}_{u,v,t} - \mathbf{O}_t)\|},$$

where $\mathbf{D}_{u,v,t}$ represents the depth at pixel (u, v) in frame t, $\mathbf{O}_t$ is the camera center of frame t, $\mathbf{p}_{u,v,t} - \mathbf{O}_t$ is the viewing direction, and f_t is the focal length at time t. The factor α is used to avoid extreme values when $\|\mathbf{n}_{k'} \cdot (\mathbf{p}_{u,v,t} - \mathbf{O}_t)\| \approx 0$.

Finally, we check if the index $\mathcal{S}^{(s)}$ already contains a surfel $\mathbf{s}_k$ similar to the new surfel $\mathbf{s}_{k'}$. Two surfels match if their centers are within a distance d and the cosine similarity between their normals is above a threshold θ. If such a surfel $\mathbf{s}_k$ is found, we add the frame index t to the surfel's set of past view indices: $\mathcal{I}_k \leftarrow \mathcal{I}_k \cup \{t\}$, and discard $\mathbf{s}_{k'}$. If not, we set $\mathcal{I}_{k'} = \{t\}$ and add $\mathbf{s}_{k'}$ to the index $\mathcal{S}^{(s+1)}$. This process transforms the surfel memory from $\mathcal{S}^{(s)}$ to $\mathcal{S}^{(s+1)}$ with $N^{(s+1)}$ surfels for the next generation step.

3.2. Novel view generator

We use a camera-conditioned image-set generator ψ to generate novel views. The generator ψ takes K retrieved reference views $\mathcal{V}^* = \{v_t = (\mathbf{x}_t, \mathbf{c}_t)\}_{t \in \mathcal{I}^*}$ and the target camera poses $\{\mathbf{c}_{T+m}\}_{m=1}^{M}$ for the M new frames to be generated and samples the novel views:

$$\{\mathbf{x}_{T+m}\}_{m=1}^{M} \sim \psi\left(\{(\mathbf{x}_t, \mathbf{c}_t)\}_{t \in \mathcal{I}^*}, \{\mathbf{c}_{T+m}\}_{m=1}^{M}\right),$$

where $\mathcal{I}^* \subseteq \{1, 2, \ldots, T\}$ are the frame indices of the retrieved reference views.

Specifically, we base our generator on the recent SEVA [46] model. Given the plug-and-play nature of our memory module, VMem can work with other image-set generators as well. We also fine-tune a more efficient version of SEVA that operates with a reduced number of reference frames. We provide more details in Sec. 4.1.

4. Experiments

We evaluate our method on established benchmarks for camera-conditioned autoregressive view generation from

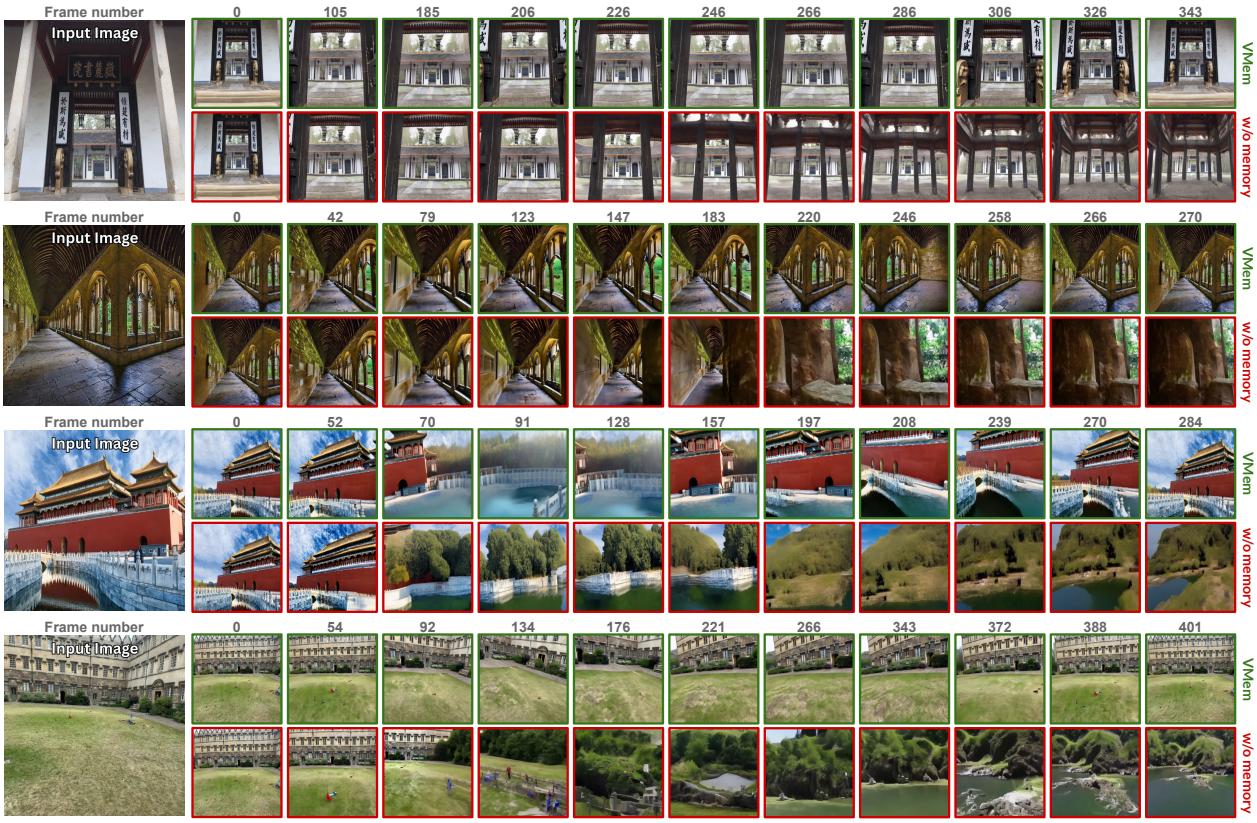


Figure 5. **Long sequences with revisitations.** We compare our VMem against a baseline without memory that relies solely on the last K frames for context. Each sequence: input images (left), then generated images at selected frames. Our method (top rows) maintains consistency when revisiting observed regions, while the baseline (bottom rows) shows severe inconsistencies across extended sequences.

	Short-term (50^{th} frame)						**Long-term** ($\geq 200^{th}$ frame)					
	LPIPS ↓	**PSNR** ↑	**SSIM** ↑	**FID** ↓	R_{dist} ↓	T_{dist} ↓	**LPIPS** ↓	**PSNR** ↑	**SSIM** ↑	**FID** ↓	R_{dist} ↓	T_{dist} ↓
GeoGPT [25]	0.444	13.35	—	26.72	—	—	0.674	9.54	—	41.87	—	—
Lookout [22]	0.378	14.43	—	28.86	1.241	0.876	0.658	10.51	—	58.12	1.142	0.924
PhotoNVS [42]	0.333	15.51	—	21.76	—	—	0.588	11.54	—	41.95	—	—
GenWarp [26]	0.436	12.03	0.144	29.69	0.564	0.059	0.613	9.56	0.085	36.40	0.850	**0.251**
MotionCtrl [35]	0.424	12.00	0.148	19.25	0.341	0.356	0.605	9.13	0.083	35.45	**0.748**	0.609
ViewCrafter [43]	0.377	16.97	0.262	25.39	1.562	0.208	0.592	9.74	0.148	34.12	2.177	0.827
SEVA [46] ($K = 17$)	0.293	18.33	0.382	17.29	0.223	0.118	0.455	13.98	0.216	23.82	1.125	0.742
VMem ($K = 4$)	**0.287**	**18.49**	**0.406**	**17.12**	**0.219**	**0.039**	0.493	13.12	0.183	27.15	0.811	0.499
VMem ($K = 17$)	0.293	18.33	0.382	17.29	0.223	0.118	**0.452**	**14.09**	**0.227**	**23.56**	0.982	0.432

Table 1. **Single-view NVS on RealEstate10K.** Frames are subsampled with 10-frame intervals. K denotes the number of context views. In the short-term setting, all past frames fit in the context windows for both SEVA [46] and VMem; hence, SEVA [46] and VMem ($K = 17$) yield identical results since VMem uses SEVA as the backbone. VMem ($K = 4$) is fine-tuned on RealEstate10K for this setting, yielding better performance than SEVA [46] and VMem ($K = 17$) in short-term scenarios. VMem surpasses baselines on most metrics in this benchmark. However, its **true advantage—spatial consistency—is not fully reflected** because RealEstate10K trajectories rarely revisit previously observed areas. We address this limitation in Sec. 4.3.

a single image, comparing our models to previous open-source approaches (Sec. 4.2). These benchmarks, however, have a critical limitation: camera trajectories rarely revisit previously observed regions, unlike real-world behavior where humans or robots often return to or look at previously explored areas. To address this limitation, we propose a cycle-trajectory evaluation protocol that extends these benchmarks by making the camera return to the starting position along the same path in reverse order (Sec. 4.3). Additionally, we collect a small dataset with trajectories that

Method	LPIPS ↓	PSNR ↑	SSIM ↑	FID ↓	R_{dist} ↓	T_{dist} ↓
Look-out [22]	0.809	8.41	0.069	38.34	1.262	0.341
GenWarp [26]	0.507	11.13	0.134	32.94	0.848	0.241
MotionCtrl [35]	0.589	9.07	0.096	26.86	0.871	0.293
ViewCrafter [43]	0.401	11.82	0.217	24.72	0.902	0.492
SEVA [46] ($K = 17$)	0.401	11.82	0.217	24.72	0.902	0.492
VMem ($K = 4$)	0.397	15.72	0.297	24.97	**0.821**	0.392
VMem ($K = 17$)	**0.304**	**18.15**	**0.377**	**24.18**	0.892	**0.165**

Table 2. **Results on cycle trajectories from RealEstate10K.**

Method	LPIPS ↓	PSNR ↑	SSIM ↑	R_{dist} ↓	T_{dist} ↓
Look-out [22]	0.792	8.52	0.008	**0.727**	1.499
GenWarp [26]	0.521	10.27	**0.129**	0.785	1.982
MotionCtrl [35]	0.692	8.21	0.082	0.842	**0.129**
ViewCrafter [43]	0.494	13.62	**0.129**	1.643	0.492
VMem ($K = 4$)	**0.472**	**14.11**	0.121	1.204	0.387

Table 3. **Results on cycle trajectories from Tanks and Temples.**

revisit observed regions for qualitative evaluation in Fig. 5. Finally, we conduct an ablation study in Sec. 4.4.

4.1. Experimental setup

Implementation details. We use pre-trained SEVA [46] as the generator ψ. SEVA uses a fixed total number of reference and target frames ($K+M = 21$). For our experiments, we use $M = 4$ target views, which leaves $K = 17$ context views. However, this configuration demands substantial computational resources that may be prohibitive for many applications. To address this limitation, we fine-tune a more efficient version with LoRA [13] on the RealEstate10K training split [47] that works with a reduced number of reference frames ($K = 4$) and target views ($M = 4$), and use this efficient version for our main experiments. We demonstrate (Secs. 4.3 and 4.4) that this fine-tuned version, when combined with VMem, achieves comparable performance to the original SEVA while delivering approximately $12\times$ speedup at inference time. We provide more technical details in the supplementary material.

Datasets. We consider two real-world datasets for novel view generation evaluation: RealEstate10K [47], comprising indoor scene video clips, and Tanks-and-Temples [14], including indoor and outdoor scenes with larger camera motions. For qualitative evaluation, we use in-the-wild images collected from the internet or captured with phone cameras as input views (see Fig. 5).

Evaluation metrics. We evaluate generated novel view quality based on three factors: (1) overall quality of generated novel views, comparing generated novel view distributions to test set distributions using Fréchet Image Distance (FID) [10]; (2) the ability of the model to preserve image details across views, measuring peak signal-to-noise ratio (PSNR) for pixel differences, along with LPIPS [45] and SSIM [34], as in [22]; and (3) alignment between generated camera poses and ground truth, following [9], where we express estimated camera poses relative to the first frame and normalize translation by the furthest frame. We use DUSt3R [33] to extract poses from generated views. We calculate rotation distance R_{dist} by comparing ground truth and extracted rotation matrices of each generated sequence:

$$R_{\text{dist}} = \arccos\left(0.5\left(\text{tr}(\mathbf{R}_{\text{gen}}\mathbf{R}_{\text{gt}}^{T}) - 1\right)\right),$$

where $\mathbf{R}$ denotes a rotation matrix. We compute translation distance t_{dist} as:

$$t_{\text{dist}} = \|\mathbf{t}_{\text{gt}} - \mathbf{t}_{\text{gen}}\|_2 .$$

Context views K	View Retrieval	LPIPS ↓	PSNR ↑	SSIM ↑	R_{dist} ↓	T_{dist} ↓
17	Temporal	0.477	13.92	0.188	0.976	0.254
	Camera Distance (SEVA [46])	0.397	15.72	0.297	**0.821**	0.392
	Field of View	0.374	15.75	0.292	0.911	0.382
	VMem	**0.304**	**18.15**	**0.377**	0.892	**0.165**
4	Temporal	0.794	7.52	0.018	1.942	0.458
	Camera Distance	0.422	13.27	0.187	1.787	0.319
	Field of View	0.424	13.11	0.192	1.782	0.285
	VMem	**0.381**	**14.82**	**0.275**	**0.793**	**0.124**

Table 4. **Ablation study on cycle trajectories from RealEstate10K.** We compare different view retrieval strategies for two context view counts K

4.2. Short- and long-term view generation

We first evaluate VMem on the benchmark from [22, 42], which defines two scenarios: *short-term* and *long-term* novel view generation. Beginning from the first frame of each ground-truth test sequence, the model autoregressively generates images along the ground truth camera trajectories with 10-frame intervals. The *short-term* scenario evaluates the fifth generated image (50 frames from the initial view), while the *long-term* scenario evaluates the final image ($\geq$200 frames from the initial view). Following the protocol, we only consider samples with more than 200 frames for the *long-term* scenario and more than 50 frames for the *short-term* scenario. We report results in Tab. 1. VMem substantially outperforms all baselines on the key metrics, demonstrating superior scene extrapolation. However, the benchmark trajectories rarely revisit observed regions, limiting evaluation of long-term scene consistency.

4.3. Long-term view generation with revisitations

To evaluate long-term scene consistency when revisiting observed regions—a scenario that rarely occurs in existing benchmarks—we propose a cycle-trajectory evaluation protocol as a proxy for real revisitation scenarios in human and robotic exploration. In this protocol, the model generates frames following trajectories from initial to final poses, then returns along the same path to the starting position.

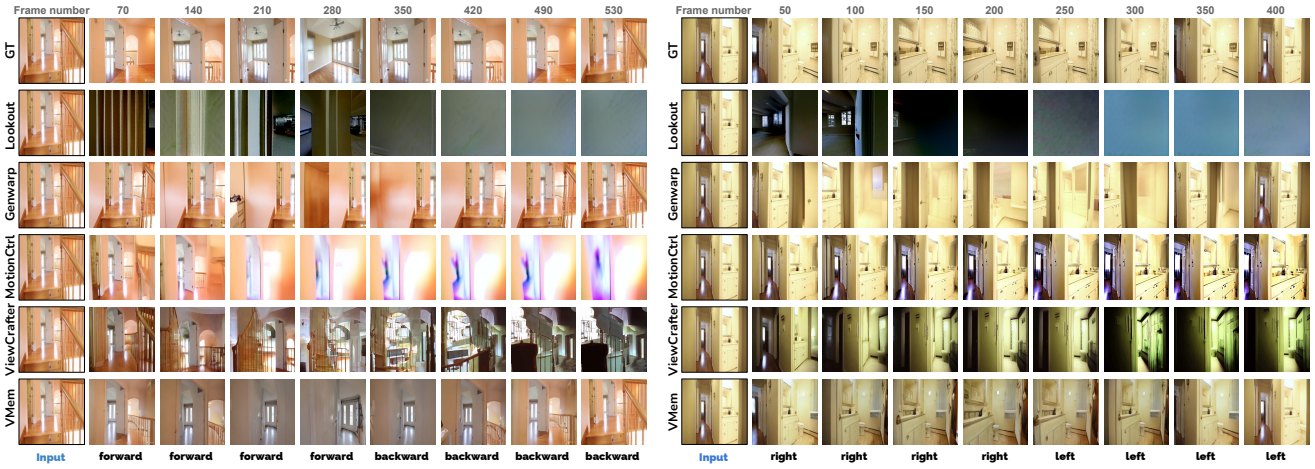


Figure 6. **Qualitative comparison of the cycle trajectory frames** ($\geq$ 400 frames). VMem autoregressively generates new views while maintaining memory of all previously generated frames, ensuring consistency when revisiting previously seen locations.

We evaluate metrics every 10 frames on the return trajectories, using all test sequences. Following this protocol, we report quantitative results in Tab. 2 and qualitative comparisons in Fig. 6. VMem outperforms all baselines across all metrics in cycle-trajectory evaluation, demonstrating superior visual consistency when revisiting previously observed regions.

To validate generalization, we evaluate on Tanks-and-Temples containing indoor and outdoor scenes with more dynamic camera trajectories. We use all six advanced scenes and apply the same *cycle-trajectory* protocol using the first 50 frames, evaluating every frame without temporal subsampling. Results are in Tab. 3. VMem consistently outperforms all baselines across in-domain (Tab. 2) and out-of-domain (Tab. 3) evaluations on the majority of metrics.

We also qualitatively evaluate VMem across diverse in-the-wild scenes in Fig. 5, demonstrating generation of diverse, consistent, high-quality long videos on in-the-wild scenarios.

4.4. Ablation study

We ablate our surfel-based memory indexing for context view retrieval in VMem. We evaluate our approach in the cycle-trajectory setting against three alternative retrieval strategies: (1) temporal-based retrieval, selecting the most recent K views from memory, (2) camera distance-based retrieval, selecting top K views with cameras closest to the target view camera—also used by SEVA [46], our backbone generator, and (3) field-of-view-based retrieval, selecting top K views with highest field-of-view overlap with the target view. We also evaluate performance trade-offs between our lightweight, fine-tuned generator using $K = 4$ context views versus original SEVA with $K = 17$ context views. We report results in Tab. 4.

Our findings show VMem consistently improves generator performance across both numbers of context views. Improvement is most pronounced with fewer context views ($K = 4$), highlighting the effectiveness of our surfel-based memory index.

Computational efficiency. A key practical advantage of our approach is its computational efficiency. VMem with only $K = 4$ views achieves $12\times$ speed improvement over the original SEVA while recovering most of the performance achieved with $K = 17$ views. On an RTX 4090 GPU, our method generates frames in 4.2 seconds compared to 50 seconds for SEVA, representing a step towards real-time interactive scene exploration.

5. Conclusion

We introduced *Surfel-Indexed View Memory (VMem)*, a novel plug-and-play memory module for long-term autoregressive scene generation using a surfel-indexed view memory. By anchoring past views to a surfel-based representation of the scene and retrieving the most relevant ones, our approach improves long-term scene consistency while reducing computational costs. Experiments on long-term scene synthesis benchmarks demonstrate that VMem outperforms existing methods in scene coherence while enabling the use of fewer context views, leading to significantly faster generation. Our work advances scalable video generation, with applications in virtual reality, gaming, and other domains.

Acknowledgments. The authors of this work are supported by Clarendon scholarship, ERC 101001212-UNION, and AIMS EP/S024050/1. The authors would like to thank Xingyi Yang, Zeren Jiang, Junlin Han, Zhongrui Gui for their insightful feedback.

References

[1] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. SF3D: Stable fast 3D mesh reconstruction with uv-unwrapping and illumination disentanglement. *arXiv preprint arXiv:2408.00653*, 2024. 2

[2] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. 3

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 2

[4] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-Stage Diffusion NeRF: A Unified Approach to 3D Generation and Reconstruction, 2023. arXiv:2304.06714 [cs].

[5] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024.

[6] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[7] Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. Scenescape: Text-driven consistent scene generation. *arXiv preprint arXiv:2302.01133*, 2023. 1, 2

[8] Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 2024. 2, 3

[9] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024. 7

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[11] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7909–7920, 2023. 1, 2

[12] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *Proc. ICLR*, 2024. 2

[13] 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. 7

[14] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 7

[15] Jing Yu Koh, Harsh Agrawal, Dhruv Batra, Richard Tucker, Austin Waters, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Simple and effective synthesis of indoor 3d scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1169–1178, 2023. 1, 2

[16] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021. 1, 2

[17] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. RealFusion: 360 reconstruction of any object from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[18] Norman Müller, Katja Schwarz, Barbara Rössle, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. Multidiff: Consistent novel view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10258–10268, 2024. 1, 2

[19] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5480–5490, 2022. 2

[20] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model, 2024. 1, 2, 3

[21] Stefan Popov, Amit Raj, Michael Krainin, Yuanzhen Li, William T. Freeman, and Michael Rubinstein. Camctrl3d: Single-image scene exploration with precise 3d camera control, 2025. 1, 2

[22] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3563–3573, 2022. 2, 3, 4, 6, 7

[23] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed

world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 3

[24] Chris Rockwell, David F Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14104–14113, 2021. 1, 2

[25] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14356–14366, 2021. 2, 3, 4, 6

[26] Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. Genwarp: Single image to novel views with semantic-preserving generative warping. *arXiv preprint arXiv:2405.17251*, 2024. 1, 2, 3, 6, 7

[27] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv.cs*, abs/2310.15110, 2023. 2

[28] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arxiv*, 2024. 2

[29] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter Image: Ultra-fast single-view 3D reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[30] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4190–4200, 2023. 2

[31] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16773–16783, 2023. 3

[32] Qianqian Wang*, Yifei Zhang*, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 5

[33] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 7

[34] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[35] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. *ACM SIGGRAPH 2024 Conference Papers*, 2023. 2, 3, 4, 6, 7

[36] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 1, 2

[37] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21551–21561, 2024. 2

[38] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 2

[39] Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory, 2025. 3

[40] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T. Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, and Charles Herrmann. Wonderjourney: Going from anywhere to everywhere. *arXiv.cs*, abs/2312.03884, 2023. 2

[41] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T. Freeman, and Jiajun Wu. WonderWorld: interactive 3D scene generation from a single image. *arXiv*, 2406.09394, 2025. 2

[42] Jason J Yu, Fereshteh Forghani, Konstantinos G Derpanis, and Marcus A Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7104, 2023. 2, 3, 4, 6, 7

[43] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 1, 2, 3, 6, 7

[44] Shangjin Zhai, Zhichao Ye, Jialin Liu, Weijian Xie, Jiaqi Hu, Zhen Peng, Hua Xue, Danpeng Chen, Xiaomeng Wang, Lei Yang, Nan Wang, Haomin Liu, and Guofeng Zhang. Stargen: A spatiotemporal autoregression framework with video diffusion model for scalable and controllable scene generation, 2025. 3

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[46] 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. 3, 5, 6, 7, 8

[47] 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. 7