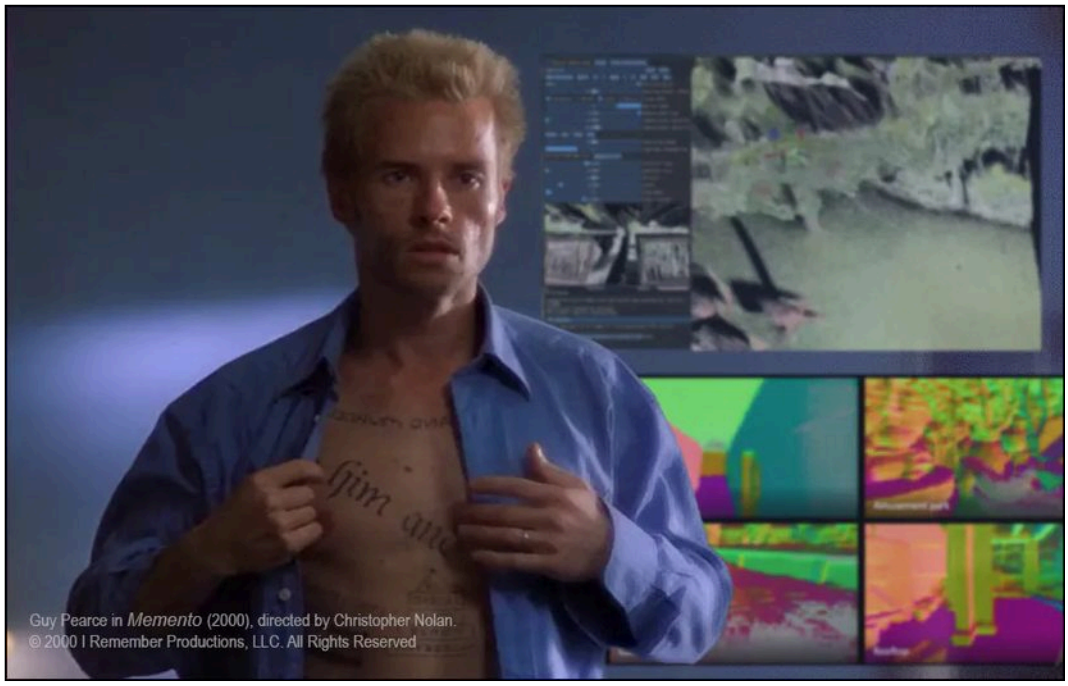
Giving AI Video Some Remembrance of Things Past

Originally published Tues 20 October 2026 at https://www.unite.ai/giving-ai-video-some-remembrance-of-things-past/



Back in March we looked at the difficulty generative AI video has in remembering things and places that it has seen, so that it can reproduce them a second time, and offer visual consistency.

The net effect of the problem is that you may walk through a door in an AI world model, turn round ten seconds later, and find that the door you came through is no longer there.

It's not a trivial glitch to fix: since generative AI video is constantly inventing 'the next thing' it must contend with, it cannot really remember anything very far back, or anything that has expired from its narrow context window.

This means that if you explore a trained AI world model, it's not like exploring a modeled CGI environment, where the textures and geometry are fixed, and saved as vector and bitmapped data; instead, it's like walking through a dream of *approximate probabilities* – you may find the exact same thing twice, but that's only because it was summoned up a second time from the latent space via context; not because the model actually remembers creating it or seeing it.

Thus, the average genAI video world model is similar to someone with anterograde amnesia, who has total recall prior to a cerebral event – such as a stroke or injury – but who cannot make new memories *since* the incident (a condition given publicity by Guy Pearce's character suffering from it in Christopher Nolan's *Memento* [2000], featured in the main article image).

In this way, video models retain a deep and profoundly-embedded understanding of many domains; but without the capability to remember anything new.

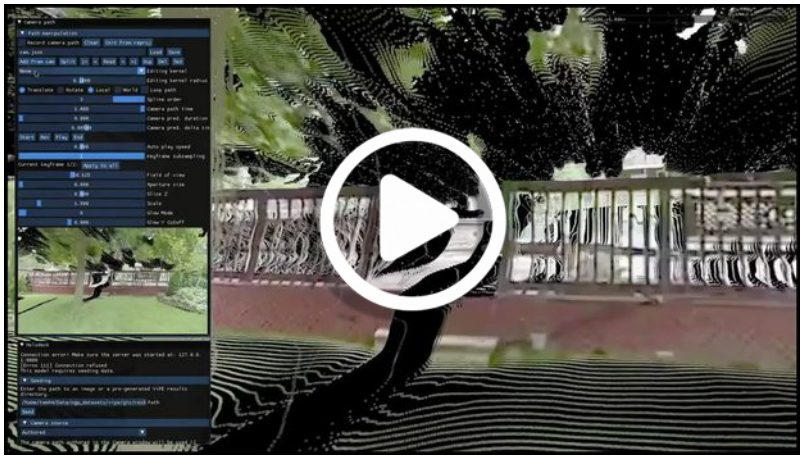
Trail of Breadcrumbs

The HyDRA solution that we looked at in March was based on the use of a 'memory bank' of compressed visual tokens that would allow AI-generated characters to move offscreen and return in the same guise as when they were first seen.

There are a number of other methods that use this 'breadcrumb' approach; however, generating and leaving behind complex 3D geometry solves the problem at the expense of file size, as well as responsiveness, latency, and other potential bottlenecks.

These '3D-ground world models' include the NVIDIA-led Gen3C, whose HyDRA approach uses the most typical method of resolving this particular problem – turning what it 'sees' into point-clouds, which are essentially stripped-down 3D geometry:

© Martin Anderson and/or commissioning entities



CLICK TO PLAY VIDEO ON SITE

Click to play. *The HyDRA approach uses a persistent 3D point-cloud cache to guide video generation, preserving scene consistency as the camera moves and previously seen areas reappear.* Source

There are other similar approaches, such as Oxford University's 2025 VMEM, and the Stanford-led SPMem collaboration from the same period:



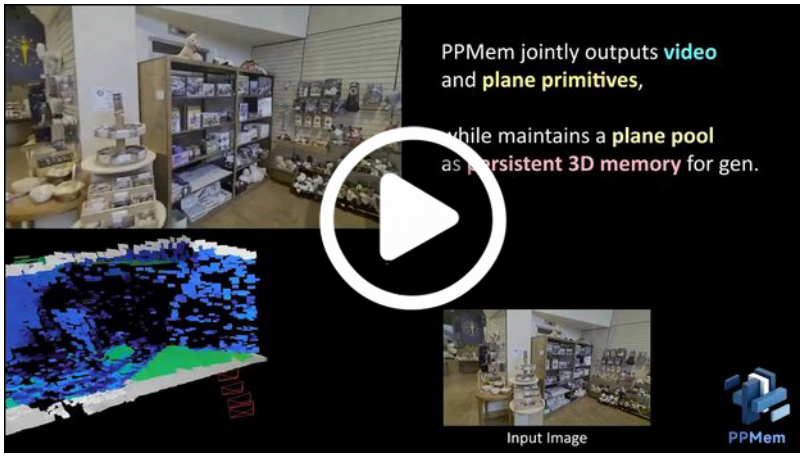
CLICK TO PLAY VIDEO ON SITE

Click to play*. From the Stanford-led collaboration SPMem, which combines a persistent 3D map of static scenery with recent video frames for motion, and selected older frames for visual detail, helping the model remember places when revisiting them.* Source

But all 3D geometry is heavy work, and hard on inference requirements. It's not easy to walk round an environment and be constantly 'building' it as a CG artist might. To boot, any other clips or viewpoints or takes will have to be able to reference the created geometry later, which is a challenging proposition.

Lighter Notes

The latest entry to take up the 'AI video memory' is a new collaboration led by Tsinghua University, titled *Persistent Planar Memory for Video World Models*:



CLICK TO PLAY VIDEO ON SITE

© Martin Anderson and/or commissioning entities

Click to play. *From the project site for the Tsinghua-led PPMem initiative, an example of the system in action. PPMem replaces the volumetric 3D memory used by the prior project SPMem with a more compact collection of flat surfaces, updating and reusing them as the camera moves to keep previously seen environments consistent.* Source

As indicated in the video segment above*, PPMem improves on the prior approaches by using a more compact representation of the 3D environment. Instead of storing millions of individual points to reconstruct previously-seen environments, the system identifies flat surfaces, such as walls, floors and furniture, and stores them as reusable geometric planes, called *plane primitives*; and these are continuously added to a persistent local memory as new parts of the environment are generated:



Two examples of PPMem generating video while building a persistent 3D memory. Each example begins with an input image on the left, with the camera's planne Reading the numbered frames from left to right across the first row, then the second, shows the camera exploring a shop or garden. The small colored insets reve accumulate as the journey progresses, preserving previously explored scenery. Source

When the camera returns to an earlier location, the stored planes are projected back into its field of view, guiding the generation of new frames, and reducing the amount of spatial information that must be retained and processed. This also helps to prevent familiar objects and surroundings from changing or disappearing between visits.

The use of plane primitives – parametric objects defined by mathematical vectors – provides a more agile and adroit resolution of the 'breadcrumb approach', the authors assert:

> *'A model that generates entire worlds, not just short clips, needs an analogous compact, structured memory of what it has produced. Plane primitives are a natural candidate for this role.*
>
> *'Each primitive is parameterized by just 7 numbers (a center depth, two rectangular extents, and a rotation quaternion), which explicitly represents 3D geometric structure, and is orders of magnitude more compact than per-pixel representations.'*

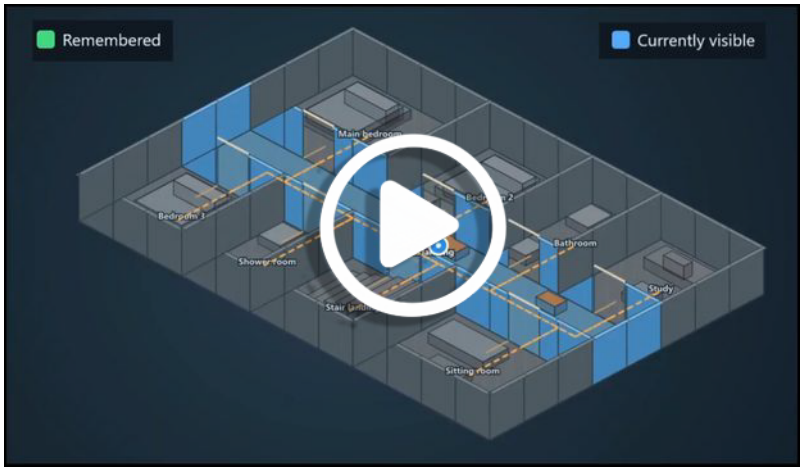


CLICK TO PLAY VIDEO ON SITE

Click to play if necessary. *Lightweight assemblies of plane primitives generated by PPMem can capture and persist detail as well as broad geometry, as the camera passes through the world model. The green sections visualized above are retained in memory as they are mapped from dark>green.*

The extent to which such planes can represent an explored world is naturally constrained by memory and other predictable bottlenecks, meaning that it would not be possible to roam the world for hours and return to find all as you left it; but then, this kind of 'level-based' restriction will be familiar to videogame enthusiasts, whose powerful but finite local resources tend to obtain the best results in the most confined conditions:

© Martin Anderson and/or commissioning entities



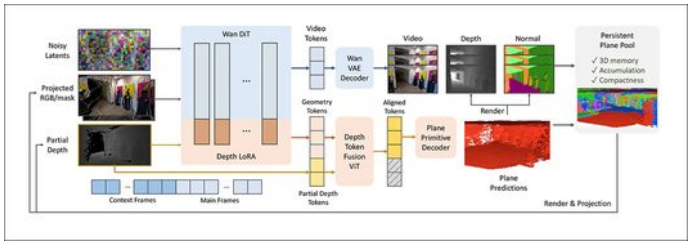
CLICK TO PLAY VIDEO ON SITE

Click to play. *In PPMem, as the camera traverses an environment, it leaves lightweight 'planes' (shown by the green areas) that represent what it saw, and which will still be there if the camera should rove back again.*

Therefore, above, we can see such a level being 'mapped', room by room, until the entire floor is green and freely explorable with less (but not zero) need to generate more planes.

Method

PPMem was developed from Wan2.2-TI2V-5B (rather than the Wan2.1-T2V-1.3B used by HyDRA), a five-billion-parameter video diffusion model, with the addition of a geometry prediction system adapted from Plana3R. The existing video backbone was retained, while a separate processing path was introduced to extract geometric information from its internal representations:



Schema for PPMem's video generation and 3D memory system. The upper path produces the video, while the lower path records the shapes and positions of visi surfaces. These are added to the stored memory on the right, which is then used to guide the next section of video. The arrows along the bottom show how this information is fed back into the system.

In the first of two training stages, Wan 2.2 was fine-tuned for camera-controlled video generation, using projected views of the source image and earlier frames to establish the intended camera movement. The video backbone was then frozen, and the new geometry components trained independently.

For training, more than 80,000 clips from DL3DV and RealEstate10K were used at a resolution of 576×1024. Reference depth information was obtained from MoGe-2, with the estimated distances aligned to 3D reconstructions produced by VGGT.

PPMem's plane-prediction system was thus trained to estimate surface depth and orientation, with output checked against reference geometry. Additional training was applied to the rendered surfaces to improve their agreement with the reconstructed scenes.

Low-Rank Adaptation (LoRA) was used to introduce geometry-specific processing, without modifying the underlying video model.

During inference, the normal video-generation process is completed with these adaptations disabled, and a separate processing pass then performed to extract the geometry, allowing the two operations to remain essentially independent.

Video is then generated in segments of approximately 25 frames, with the intended camera movements used to prepare the visual and spatial information needed for each new sequence. This includes projected views of previously- generated scenery; masks indicating which areas are visible; and depth information derived from the accumulated geometry.

To improve the prediction of new surfaces, stored depth information is then combined with the model's existing understanding of the scene, using a visual transformer based on DINOv2-Base.

During training, portions of the depth maps were held back through selective masking, requiring the model to predict surface geometry from incomplete spatial data and improving generalization.

With new geometry generated, the plane pool is refined through checks for overlapping and obscured surfaces, with less reliable estimates removed when better observations become available. This allows the stored representation to be *progressively corrected* as the camera moves, rather than simply accumulating every surface prediction indefinitely.

© Martin Anderson and/or commissioning entities

Results

For tests, the method was trialed against Gen3C; Voyager; VMEM; and SPMem, all on their default repo code and parameters.

Metrics used were Peak Signal-to-Noise Ratio (PSNR); Learned Perceptual Similarity Metrics (LPIPS); CLIP similarity; and DINOv2 similarity (see paper for detailed exceptions).

In the first test, PPMem was compared against the contenders on the DL3DV and RealEstate10K datasets, with video quality evaluated over sequences of 49 and 121 frames:

		$N=49$				$N=121$			
Dataset	Method	PSNR↑	LPIPS↓	$CLIP_{sim}$ ↑	$DINOv2_{sim}$ ↑	PSNR↑	LPIPS↓	$CLIP_{sim}$ ↑	$DINOv2_{sim}$ ↑
DL3DV	Gen3C [24]	15.74	0.474	0.847	0.710	13.59	0.565	0.798	0.585
	Voyager [11]	12.84	0.568	0.796	0.615	11.35	0.649	0.760	0.516
	VMem [13]	11.88	0.636	0.781	0.477	10.87	0.722	0.696	0.274
	SPMem [37]	14.70	0.492	0.853	0.703	13.03	0.607	0.788	0.536
	PPMem (ours)	**16.99**	**0.335**	**0.909**	**0.861**	**13.98**	**0.464**	**0.863**	**0.769**
RE10K	Gen3C [24]	21.93	0.238	0.925	**0.949**	19.22	0.303	0.912	0.906
	Voyager [11]	17.43	0.315	0.900	0.877	15.88	0.389	0.881	0.827
	VMem [13]	11.69	0.626	0.812	0.547	11.07	0.678	0.754	0.402
	SPMem [37]	18.42	0.279	0.920	0.902	16.43	0.375	0.892	0.819
	PPMem (ours)	**22.46**	**0.168**	**0.965**	**0.949**	**19.34**	**0.254**	**0.952**	**0.910**

Video quality comparison of PPMem, Gen3C, Voyager, VMem and SPMem on DL3DV and RealEstate10K, over 49 and 121 frames. PPMem achieved the best or joint-best scores across all four metrics, with its advantage generally increasing over longer sequences.

Of these results, the authors state:

> *'PPMem achieves the best results across all metrics on both datasets and both frame budgets, despite using a smaller base model (Wan 2.2 5B) compared to Gen3C (Cosmos 7B) and Voyager (HunyuanVideo ~13B). T'*

PPMem was also tested for its ability to estimate the distances of objects and surfaces from the camera: across 121 frames from DL3DV, the authors note, more accurate depth estimates were obtained than with Voyager, the only competing model capable of generating video and depth information together.

The second, qualitative test evaluated the ability of the contenders to maintain consistent scenery as the camera moved through DL3DV scenes over 300–400 frames was compared. PPMem preserved the appearance and structure of previously seen surroundings more successfully than the competing models, with its advantage becoming greater as the sequences continued:



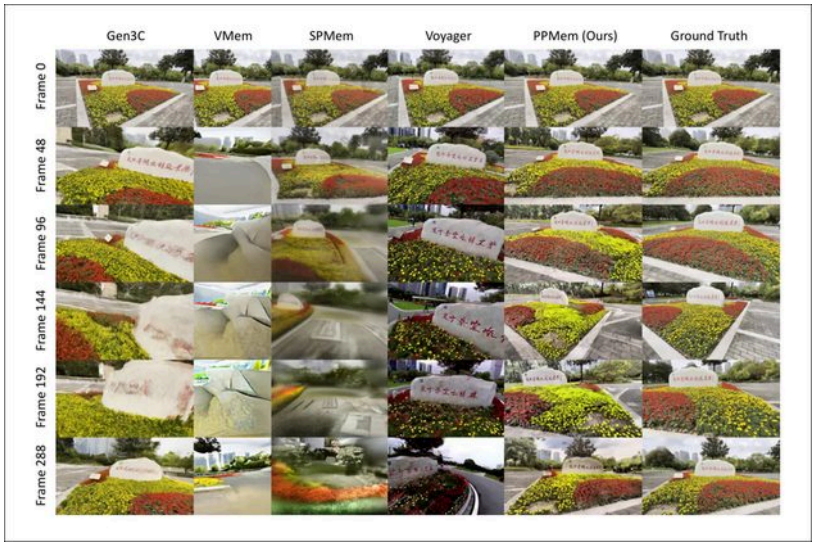
CLICK TO PLAY VIDEO ON SITE

Click to play. *Though we are only able to reproduce the results in lower resolution than the original, we felt that video was a better illustration than the static results from the paper, and refer the reader to the better-quality version at the PPMem project site.*

Here the authors state:

> *'PPMem maintains the highest perceptual fidelity throughout, with the gap widening over time — most dramatically on DINOv2, where PPMem leads the next-best method by ~0.27 at frame 300.'*

© Martin Anderson and/or commissioning entities



Comparison of video frames generated by Gen3C, VMem, SPMem, Voyager and PPMem at six points across a 288-frame sequence, alongside the original footag preserves the flowerbed's appearance and surrounding scenery more consistently as the camera moves, while competing methods increasingly distort or lose det

The paper notes that after around 400 frames of video, an entire explored scene can be represented in PPMem by approximately 100,000 plane primitives, *occupying just 5MB of memory*.

Since this is roughly a thousand times more compact than storing the same scenery as individual points in a dense 3D point cloud, a paper that might have been considered an incremental advance on an established thread in the literature apparently offers a substantive improvement in this architecture's logistics.

Conclusion

Though the new paper is an exciting development within the confines and parameters of the target problem, we are still laboring in a paradigm wherein random exploration of the latent space of a world model *sprays geometry around like persistent foam*; geometry that may need to be recovered by later scenes or for later contexts.

That recovery operation is far messier than ingesting traditional CGI meshes and wrapped textures. This entire approach could be argued as 'somewhat backward': AI models should ideally be able to retrieve and reference fixed assets through some form of RAG or lookup system, while retaining the flexibility and creative freedom of moving through latent space.

It's a lot to ask, but the 'breadcrumb' method does seem to me to be the equivalent to the bizarre solutions that preceded the Wright brothers.

* *Please see the project site for the full video.*

First published Tuesday, October 20, 2026

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Website: martinanderson.ai
Contact: martin@martinanderson.ai