

GimbalDiffusion: Gravity-Aware Camera Control for Video Generation

Frédéric Fortier-Chouinard¹, Yannick Hold-Geoffroy², Valentin Deschaintre²,
Matheus Gadelha², Jean-François Lalonde¹

¹ Université Laval, Québec, Canada

² Adobe, San Jose, USA

frederic.fortier-chouinard.1@ulaval.ca

<https://lvsn.github.io/GimbalDiffusion>



Fig. 1: We propose GIMBALDIFFUSION, a framework for absolute camera control in text-to-video generation. Our approach adapts foundational video generation models to accept absolute camera controls, conditioning the entire video on camera parameters expressed in a gravity-aligned global coordinate system. This enables the generation of videos with control over translation and extreme camera angles—such as very high pitch (top) or roll (bottom)—directly from text, something existing methods struggle to achieve. Here, the results were obtained with GIMBALDIFFUSION trained on WAN-2.2 5B with Plücker conditioning. Refer to the supplementary material for the full videos.

Abstract. Recent progress in text-to-video generation has achieved remarkable realism, yet fine-grained control over camera motion and orientation remains elusive, especially with extreme trajectories (e.g., a 180° turnaround, or looking directly up or down). Existing approaches typically encode camera trajectories using relative or ambiguous representations, limiting precise geometric control and offering limited support for large rotations. We introduce GIMBALDIFFUSION, a framework that enables camera control grounded in physical-world coordinates, using gravity as a global reference. Instead of describing motion relative to previous frames, our method defines camera trajectories in an absolute coordinate system, allowing accurate, interpretable control over camera parameters. Using panoramic 360° videos for training, we cover the full sphere of possible viewpoints, including combinations of extreme pitch and roll that are out-of-distribution of conventional video data. To improve camera control, we introduce null-pitch conditioning, a strategy that prevents the

model from overriding camera specifications in the presence of conflicting prompt content (e.g., generating grass while the camera points toward the sky). Finally, we propose new benchmarks to evaluate gravity-aware camera-controlled video generation, assessing models’ ability to generate extreme camera angles and quantify their input prompt entanglement.

Keywords: Video generation · Diffusion models · Camera control

1 Introduction

Text-to-video generation is rapidly advancing, offering increasingly photorealistic results and new creative workflows. Yet, the ability to control generated video content—especially camera trajectory—remains limited, particularly for extreme camera rotations such as top-down shots, Dutch angles, or full 360° pans. Early approaches rely solely on textual input which does not provide fine-grained control, such as specifying geometric parameters for camera paths and angles.

Recent methods [2–4, 12, 13, 25, 37, 53] address this limitation by conditioning video models with explicit camera trajectories, frequently encoded as Plücker rays or relative camera positional encodings. However, these representations are inherently ambiguous and lack the precision of real camera systems. Typically, camera extrinsics used for Plücker conditioning are defined *relative* to the first frame, making it impossible to explicitly control camera orientation with an absolute reference within the environment (e.g., with respect to gravity). Furthermore, they are trained on videos with limited camera movement and fail to generalize to extreme camera rotations.

In this work, we introduce GIMBALDIFFUSION, a novel gravity-centric approach to data annotation that overcomes this challenge and enables precise, absolute user control over generated camera trajectories, as illustrated in Fig. 1. The name reflects the role of a gimbal—a device that stabilizes orientation in physical space—mirroring our use of gravity as a global reference to stabilize and disambiguate camera rotation. To resolve rotational ambiguity, we leverage geometrically calibrated 360° panoramic videos. By sampling camera crops over the sphere, we can remove camera bias in human-captured datasets, enabling the sampling of rare cinematic camera shots, such as barrel rolls (a “twisting” camera rotation), extreme low- and high-angle perspectives, or looping 360° rotations. The effectiveness of this approach is demonstrated in Fig. 2, where our method produces frames whose camera views more closely match the intended trajectories than those of previous methods. This enables models trained with our pipeline to learn a wide range of camera trajectories, far beyond the predominantly straight, forward-facing paths in conventional videos.

We further observe that text and camera angles are inherently entangled, and that care must be taken to ensure that appropriate control is achieved. For example, when asked to generate a video of a “green field under a blue sky” with a camera looking *down*, a model must learn to render *only* grass, even if other concepts are specified in the prompt. To this end, we propose a null-pitch

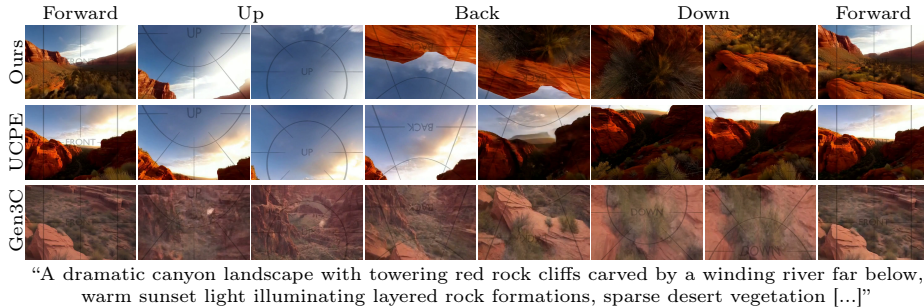


Fig. 2: Extreme rotation handling. Here, the camera starts forward and performs a motion akin to a “backflip”: looking straight up, then backwards, down, and back to the starting position. Our method (top) precisely follows the desired camera trajectory while accurately reproducing the original content from the first frame. Other methods, like UCPE [53] and GEN3C [32] (initialized with PreciseCam [5]), fail at this task. Refer to the supp. for the full videos.

conditioning approach in which captioning is performed on forward-facing crops rather than on the actual camera point of view.

Finally, we propose two new evaluation benchmarks. The first evaluates video generation with extreme camera angles by leveraging the large-scale, high-quality dataset SpatialVID-HQ [38] and rebalancing the camera pitch distribution to obtain diverse test samples. The second evaluates entanglement between prompt and camera angle by annotating crops of panoramas from the PolyHaven dataset [31]. We will release these test datasets upon publication.

In summary, we make the following contributions:

- A pipeline for generating diverse camera trajectories from panoramic 360° video, providing exhaustive coverage of possible camera orientations;
- A strategy named *null-pitch conditioning* for disentangling text from absolute camera angle, critical for extreme viewpoints where prompt content and camera orientation diverge most;
- Two evaluation benchmarks for: video generation with extreme camera angles, and assessing the entanglement between input prompt and camera orientation.

2 Related Work

Video generation. Generative video modeling has evolved rapidly, from early GAN-based methods that produced short, low-resolution clips [1, 26, 35] to large-scale diffusion- and flow-based architectures capable of generating temporally coherent, semantically consistent videos [43]. Recent open-source models such as CogVideoX [49] and WAN [37] push the frontier of video length, resolution, and controllability while proposing both text-to-video and image-to-video variants.

Editing videos remains challenging: precise changes to content, viewpoint, or lighting are difficult once the frames have been synthesized. To address this,

control strategies have emerged, introducing user-guided conditioning based on edge maps, depth, sketches, or explicit 3D geometry for structural control [6, 7, 15, 50], and illumination-aware guidance for lighting control [10, 22, 28]. These mechanisms allow precise control over motion, composition, and scene appearance, enhancing the usability and compositional fidelity of generative video systems.

Camera control methods. In static image generation, PreciseCam [5] (similarly, [27]) provide text-to-image models with absolute control over camera pose by conditioning on roll, pitch, field of view, and lens distortion alongside textual prompts. For image-to-video generation, ViewCrafter [52] and GEN3C [32] generate novel views from a single input image, but cannot handle scene motion. Methods such as MotionCanvas [48], Diffusion-as-Shader [11], and TrajectoryCrafter [51] maintain spatial coherence with point cloud conditioning from an initial reference image. While effective, these approaches depend on the reference frame’s depth and pose estimation quality, limiting flexibility and generalization.

In text-to-video synthesis, AC3D [2] allows user-defined camera trajectories but conditions motion relative to the first frame only, leaving global rotational and translational ambiguities unresolved. Similarly, Li *et al.* [25] manipulate positional encodings to control viewpoint changes, yet still lack a consistent absolute reference such as gravity. Concurrent with our work, UCPE [53] extends positional encodings to encode a latitude up map. However, its data generation is complex and requires matching 360° videos with rotations extracted from a secondary dataset. It also cannot handle extreme camera angles (see Fig. 2).

These works underscore the need for precise, absolute camera control that remains robust under extreme rotations. Our approach addresses this gap by aligning camera orientation with gravity and anchoring motion within a consistent world frame, allowing extreme camera rotations in text-to-video synthesis.

Panorama video generation methods [40, 46, 54] adapt video generation backbones for the task of generating 360° equirectangular videos. In addition to text, they can be conditioned on an input image [40, 54], optical flow [40] and derotated flow [54]. While gravity-aligned camera control can be achieved by cropping the desired field of view out of the generated panorama, this comes with several disadvantages: 1) most of the panorama pixels are discarded, resulting in severe resolution degradation; 2) the desired concepts described in the prompt can be cropped out and thus not appear in the video; and 3) current text-conditioned methods [40, 46] do not provide control over camera translation. We demonstrate these downsides in the supplementary material.

Camera pose estimation. Estimating accurate camera poses from images is a long-standing problem in 3D vision. Classical structure-from-motion pipelines such as VisualSFM [45], OpenMVG [30], and COLMAP [33] reconstruct camera intrinsics and extrinsics through feature matching, triangulation, and global bundle adjustment. More recent learning-based approaches jointly infer scene geometry and pose from data: VGGT [39] leverages transformer-based global geometry tokens for pose regression, while DUST3R [42] and MAST3R [24] predict dense

correspondence fields to recover metrically scaled 3D structure. MegaSAM [17] extends these ideas to foundation scale, achieving state-of-the-art alignment accuracy. ViPE [16] extends these ideas to handle in-the-wild videos.

Despite their accuracy, these systems typically operate in arbitrary coordinate frames and lack physical grounding in gravity or a global world reference, limiting their absolute interpretability. To mitigate this, several works estimate the gravity direction from single images [18, 36, 47]. However, acquiring datasets with gravity annotations often requires auxiliary sensors, such as accelerometers, or additional semantic information about the scene. Examples include EDINA [9], Horizon Lines in the Wild [44], and the KITTI Horizon dataset [21]. ScanNet [8] proposes a dataset of 3D scenes captured using an RGB-D sensor, but exhibits limited diversity, primarily focusing on small indoor environments.

To circumvent such constraints, we geometrically calibrate 360° videos from the PanoVid dataset [46]. By combining the up vector derived from the panoramic video with SfM-estimated intrinsics and extrinsics, we obtain a consistent absolute reference for conditioning video models on gravity-aware camera control.

3 Method

We aim to train a camera-conditioned model that supports extreme camera rotations, including full 360° trajectories, extreme pitch, and large roll, while maintaining precise control. To achieve this, we ground camera poses in an absolute, gravity-aligned coordinate system. We refer to this global gravity reference as an *absolute* reference, in contrast to most methods that describe camera motion *relative* to a previous frame. Because our absolute reference is aligned with gravity, pitch and roll are well defined, whereas yaw remains unconstrained. Throughout the paper, *absolute rotation* therefore denotes pitch and roll expressed in this shared gravity-aligned reference.

However, no large-scale video dataset provides the gravity annotations we need to train such a model. This section outlines our pipeline for obtaining such data, summarized in Fig. 3. We first extract camera pose annotations from a dataset of 360° panoramic videos (Sec. 3.2 and Fig. 3, left). Next, we sample a camera trajectory (Sec. 3.2 and Fig. 3, center-bottom). Finally, we combine the two previous steps to produce our dataset. We repeat the same procedure for an additional version of each trajectory with pitch and roll set to 0° (Sec. 3.3, Fig. 3, center-top). Before detailing the full pipeline, Sec. 3.1 explains the standard representation to encode camera poses in neural-network-compatible buffers.

3.1 Representing cameras

GIMBALDIFFUSION requires a gravity-aligned camera extrinsic matrix $\mathbf{E}_{\text{abs},f} = [\mathbf{R}_{\text{abs},f} | \mathbf{t}_f]$ and intrinsic matrix $\mathbf{K}_f$ as input. Our method is agnostic to the specific camera encoding strategy: here, we showcase how it can be applied to training models conditioned on either UCPE [53] or Plücker rays (e.g., as in [2]) encodings. Unless otherwise noted, we use the UCPE encoding by default, which better maintains translation accuracy experimentally.

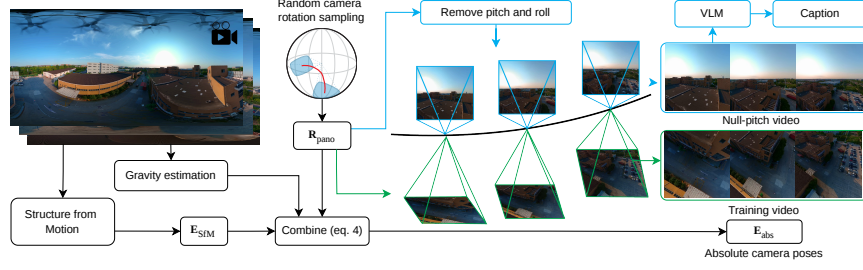


Fig. 3: Training data pipeline. On the left, we start by estimating the camera poses and gravity up vectors from a dataset of 360° panoramic videos in equirectangular format (see Sec. 3.2). We then randomly sample a camera rotation trajectory and camera intrinsics on the sphere, which are used for projecting the panoramic frames to perspective images. We integrate the two previous steps by combining the sampled rotation with the estimated camera poses (containing also translations). This process creates a dataset of perspective images (in green) with their absolute camera poses. The main advantage of this approach is the ability to sample camera viewing directions across the entire sphere, in contrast to using natural human-captured videos, which are heavily biased towards null roll and pitch, and minimal rotation amplitude. We also project the panorama on the rotation trajectory with pitch and roll removed (in blue), yielding the null-pitch video from which we obtain a textual description (see Sec. 3.3).

Option 1: UCPE encoding. Recent work [23, 25, 29, 53] has shown that camera intrinsics and extrinsics can be encoded for each token in a transformer, building upon the idea of Rotary Positional Embedding (RoPE) [34]. UCPE [53] injects camera rays through lightweight trainable self-attention layers. For each token k , a block-diagonal matrix $\mathbf{D}_k^{\text{UCPE}}$ encodes its associated 3D ray transformation:

$$\mathbf{D}_k^{\text{UCPE}} = \text{blkdiag}(\mathbf{D}_k^{\text{Ray}}, \mathbf{D}_k^{\text{RoPE}}), \quad (1)$$

where $\mathbf{D}_k^{\text{Ray}} = \mathbf{I}_{\frac{d}{8}} \otimes \mathbf{T}_k^{rw} \in \mathbb{R}^{\frac{d}{2} \times \frac{d}{2}}$ encodes the token’s ray-to-world transformation as defined in [53], and $\mathbf{D}_k^{\text{RoPE}}$ is the RoPE rotation matrix, determined by the token’s position in time and 2D image space. The $\mathbf{D}_k^{\text{UCPE}}$ transformations are then applied in the self-attention following [29]:

$$\mathbf{O} = \mathbf{D} \odot \text{Attn}(\mathbf{D}^\top \odot \mathbf{Q}, \mathbf{D}^{-1} \odot \mathbf{K}, \mathbf{D}^{-1} \odot \mathbf{V}), \quad (2)$$

where $\odot$ represents the token-wise matrix-vector product. Furthermore, unlike PProPE [25], which is relative, UCPE encodes the latitude and up-vector through a linear layer and adds them to the self-attention input.

Option 2: Plücker rays. Another common strategy for conditioning video generation models on viewpoint is to use Plücker rays [2, 3, 12, 13, 19]. For a conditioning frame f , we can define a per-pixel map of Plücker rays as $\mathbf{p}_{f,u,v} = (\mathbf{t}_f \times \mathbf{d}'_{f,u,v}, \mathbf{d}'_{f,u,v})$, where $\mathbf{t}_f$ denotes the camera origin in world coordinates, and $\mathbf{d}_{f,u,v}$ is the direction vector from the camera center toward the center of

pixel (u, v) . Each pixel is thus represented as a 6D vector describing a ray in 3D space. The ray direction can be computed as $\mathbf{d}_{f,u,v} = \mathbf{R}_f \mathbf{K}_f^{-1} [u, v, 1]^T$, with $\mathbf{d}'_{f,u,v}$ denoting the normalized direction vector $\mathbf{d}_{f,u,v}$.

3.2 Creating varied camera paths from 360° videos

Most video datasets exhibit a significant bias toward forward-looking views with horizon level or near the center of the image, as shown in the supp. material. However, such “normal” viewpoints limit storytelling expressivity by lacking cinematic shots, such as top-down, low-angle, or barrel-roll shots. Consequently, generative video models trained on such data struggle to produce these types of cinematic shots, as they lie outside the training distribution (see Fig. 2). To address this limitation, we propose generating synthetic camera trajectories by sampling and rectifying 360° videos.

Camera poses from SfM. From a 360° video, we compute the *relative* motion between frames by extracting six perspective crops (front, back, left, right, top, bottom), and processing the resulting video with a pose estimation method [16]. This yields a set of camera poses $\mathbf{E}_{\text{SfM},f}$ for each frame f . Taking the first frame of the sequence $f = 0$ as reference, we can then compute the relative motion

$$\mathbf{E}_{\text{rel},f} = \mathbf{E}_{\text{SfM},0}^{-1} \mathbf{E}_{\text{SfM},f} . \quad (3)$$

Gravity annotation. Pose estimation methods such as [16] provide *relative* estimates of camera rotations. To anchor the videos within a shared global reference frame, we align the relative camera rotations with gravity. Specifically, we use [18] to estimate the camera roll and pitch angles at frame 0. Because it operates on perspective images, we first project the panoramic frame into eight perspective views evenly sampled about the yaw axis at fixed 90° field of view. We then estimate the camera up vectors from these views and rotate them back to the null yaw reference frame. We then average the up vectors across these views, yielding $\tilde{\mathbf{u}}_0$. From this, we derive the absolute, gravity-aligned camera pose for each frame as

$$\begin{aligned} \mathbf{t}_{\text{abs},f} &= \psi(\tilde{\mathbf{u}}_0) \mathbf{t}_{\text{rel},f} \\ \mathbf{R}_{\text{abs},f} &= \mathbf{R}_{\text{pano},f} \psi(\tilde{\mathbf{u}}_0) \mathbf{R}_{\text{rel},f} \\ \mathbf{E}_{\text{abs},f} &= \varphi(\mathbf{R}_{\text{pano},0}) [\mathbf{R}_{\text{abs},f} | \mathbf{t}_{\text{abs},f}] , \end{aligned} \quad (4)$$

where $\varphi(\cdot)$ is a function that preserves only the roll and pitch angles (removes the yaw) of the first sampled camera pose, and $\psi(\cdot)$ aligns camera poses to the gravity up vector. After this adjustment, the transformed pose $\mathbf{E}_{\text{abs},f}$ has no rotation about the global up-vector axis. Please refer to the supplementary material for the full definition of $\varphi(\cdot)$ and $\psi(\cdot)$. This process establishes a gravity-aligned coordinate system consistent across all videos, enabling rotations to be defined within a common global frame. Consequently, each sequence is initialized with null yaw, reducing ambiguity and facilitating training with this representation.



Fig. 4: Training data samples from our data augmentation pipeline, capturing a highly diverse set of rotation trajectories from 360° videos. Both the trajectory and the prompt are generated on-the-fly in our dataloader. See the supp. for full videos.

Sampling random camera paths. We can generate several perspective videos from a 360° sequence by sampling random camera rotation paths. We outline our approach in Algorithm 1. We first sample random pitch, roll, and yaw angles uniformly to obtain an initial viewpoint. Next, 1 to 3 additional random rotations are sampled and distributed across the sequence length. Those rotations are applied sequentially to produce smooth and varied motion paths.

We further introduce variations in the camera field of view (FoV). For each sequence, we sample either 1, 2, or 3 keyframes at random timestamps, with an FoV ranging from 35° to 100°. The FoV values are then interpolated using cubic splines with randomized derivative boundary conditions. This approach provides much broader coverage of possible viewpoints than existing datasets (see supplementary). Notably, this sampling slightly biases viewpoints towards the poles, which we found to better support model stability and representation of more extreme viewpoints. Examples from our sampling are shown in Fig. 4.

Algorithm 1 Random camera rotation path sampling

```

pitch0 ~ Uniform(−90°, 90°) ; roll0 ~ Uniform(−90°, 90°) ; yaw0 ~ Uniform(0°, 360°)
Rpano,f ← (RYXZ(yaw0, pitch0, roll0))f=0F−1           ▷ Initialize the camera orientation
N ~ {0, 1, 2, 3}                                           ▷ Random number of rotation axes
for i = 1 to N do
    dt ~ Uniform(0, 1)                                     ▷ Random rotation duration
    ts ~ Uniform(0, 1 − dt) ; te ← ts + dt             ▷ Random start and end time
    a ~ S2                                                 ▷ Random unit axis of rotation
    θmax ~ Beta(1.0, 5.0) · 720° · dt                     ▷ Random angular displacement
    Construct cubic spline s(t) on [ts, te] with random boundary derivatives
    for f = 0 to F − 1 do                                   ▷ Loop over the F frames
        θf ← clamp(s(f/F), 0, 1) θmax                    ▷ Compute the rotation angle
        Rpano,f ← exp([θf a]×) × Rpano,f                 ▷ Update the camera orientation
    end for
end for
return Rpano

```

3.3 Null-pitch conditioning

We observed that captioning videos using their actual *rotated* crops causes the model to overemphasize the textual prompt when inferring the camera angles, rather than strictly adhering to the camera conditioning. This entanglement hinders the achievement of extreme camera angles, where visual content diverges most from the prompt. For instance, if a prompt describes ground features such as grass but the camera should face the sky, the model often still outputs a downward-looking view. To mitigate this entanglement, we caption each video using an upright, forward-looking image with a 90° field of view, thereby strengthening the model’s adherence to the camera conditioning. We refer to this strategy as “null-pitch conditioning”. Specifically, we generate two sets of crops (see Fig. 3): one using the original randomly sampled camera trajectory, and another with pitch and roll set to 0° (with yaw unchanged). As illustrated in Fig. 3, a trajectory that consistently points towards the ground would otherwise yield captions describing only ground features. A model trained on such captions would then expect these elements to appear in its outputs, even when the camera conditioning specifies a view of the sky, leading to semantically incorrect generations. Training with null-pitch captions, which provide descriptions of the full scene, makes it easier for the model to handle extreme camera angles. During training, the randomly sampled perspective crops are used for the flow or diffusion loss, while the null-pitch crops are used for captioning.

While this strategy helps the video model in handling extreme angles, we found it still struggled with undesirable elements present directly below the camera in real 360° training videos, such as tripods or the hand of a person holding a selfie stick. Indeed, since these elements are never present in the training caption, the model learns to generate these objects as if they were normal scene content. To prevent this, we generate a third set of videos (not shown in Fig. 3), this time looking straight down (-90° pitch), and caption them with the VLM. We append the resulting “look-down” caption to the null-pitch caption 50% of the time. At inference time, this “look-down” prompt is used in the classifier-free guidance as a negative prompt to ensure that undesirable content is absent from the generated videos. We found this strategy to be most useful when fully finetuning a model, which tends to lead to overfitting on unwanted artifacts present in the training set. Please see the supplementary for additional details.

4 Experiments

This section presents the training data, details model training and losses, introduces our two novel evaluation benchmarks, presents baselines, and shows quantitative and qualitative experimental results on our benchmarks.

4.1 Training data

We use the “YouTube videos” subset of the PanoVid dataset [46], which contains a diverse set of high-quality 360° videos, and filter out videos where the shortest side

is less than 900 pixels. We obtain relative camera poses and per-frame up-vectors using the procedure from Sec. 3.2. We discard videos where pose estimation either failed or resulted in extreme mean acceleration values, typically indicating unreliable camera poses. Our filtering strategy reduces the number of training videos from 7797 to 4798 with high-quality camera poses (38% discarded).

During training, we sample a random camera trajectory of 49 frames using Algorithm 1 at a resolution of 720×480 (or a similar resolution compatible with the backbone, see supp.), effectively acting as an infinite data augmentation strategy. We use InternVL-3-2B [56] for captioning, providing it with 6 evenly spaced frames across the video timeline. This is also done during training since new videos from different trajectories are sampled each iteration. Prompt lengths are further sampled between short, medium and long.

4.2 Models and baselines

We compare GIMBALDIFFUSION against recent diffusion- and flow-based methods for video generation with camera control. Since our method is invariant to the choice of architecture and can be applied to different camera conditioning mechanisms (e.g., UCPE and Plücker), we also train our method for each video generation baseline—except GEN3C, since it does not support dynamic scenes. We provide additional training and architecture details in the supplementary.

AC3D [2]. A camera encoder is trained to encode Plücker rays and inject them into the video DiT backbone. A parallel copy of 8 layers of the backbone is also trained, in a ControlNet fashion. The ControlNet is only applied for the first 40% of denoising steps. Since AC3D is trained on camera poses relative to the first frame, we provide the absolute pitch and roll information through text (see supp. for the prompt template). We also test AC3D by feeding it with absolute camera poses (“+abs. Plücker”), even though this is outside its training distribution.

We compare the original checkpoint from [2] trained on CogVideoX-2B by training GIMBALDIFFUSION in a similar fashion along with our contributions. In our case, since the model is trained on absolute Plücker rays, we do not concatenate the absolute camera information through text.

PreciseCam+WAN [37]. A straightforward approach to camera-controlled video generation is to first use a camera-conditioned image generator to synthesize the initial frame, and then apply an image-to-video model to produce the full sequence. This baseline follows that strategy: PreciseCam [5] generates the first frame conditioned on the absolute pitch, roll, and field of view, and then Wan2.1-Fun-V1.1-1.3B-Control-Camera [37] prolongs that first frame to a video. The video model is a fully finetuned checkpoint of WAN-2.1 1.3B, with an additional Plücker camera encoder and first frame input.

For GIMBALDIFFUSION, we train two variants, one using the same WAN-2.1 1.3B backbone and the other WAN-2.2 5B. For the former, we keep the original camera encoder architecture, whereas for the latter, we use a smaller camera encoder to reduce the weight count.

PreciseCam+GEN3C [32]. GEN3C [32] takes as input a 3D point cloud from a single image or an input video which can then be rendered from novel camera views. We reuse the original implementation which predicts the depth using MoGE [41] and feed it the first frame from PreciseCam [5]. We then project the point cloud along the relative camera poses and feed them to GEN3C. Note however that GEN3C can only generate novel views of static scenes, and therefore cannot produce dynamic video content (e.g., walking people).

UCPE [53] is a concurrent method that also generates videos in a gravity-aligned reference frame, using WAN-2.1 1.3B as the backbone. It injects both absolute and relative camera orientation through additional self-attention layers. Unlike ours, UCPE requires a limited auxiliary dataset to simulate rotation diversity within panoramas, resulting in a less diverse dataset than our proposed approach.

For GIMBALDIFFUSION, we train the same backbone and conditioning strategy combined with our contributions.

4.3 Video camera control benchmark

There exists no standard evaluation dataset containing paired text prompts, camera poses, and reference videos that provide extreme camera angles. Most methods [2, 13] use test splits from video datasets like RealEstate10K [55], which have very limited diversity in terms of pitch and roll angles. Please refer to the supplementary material for a diversity analysis and quantitative results on RealEstate10K.

We propose a new evaluation benchmark, featuring diverse, high-quality videos, distributed with near-uniform pitch and roll coverage. We start with 371K video clips from the “HQ” subset of the SpatialVID [38] dataset, which still exhibit a bias toward forward-looking orientations. We estimate the average pitch from the videos using Perspective Fields [18] and organize them into 10-degree bins spanning -85° to $+85^\circ$. We manually filter out videos where the pitch estimation clearly failed until we obtain a subset of 138 videos with a uniform pitch distribution. For the roll, we found that even among the 371K videos, nearly all have near-zero roll. We thus artificially add a roll angle by randomly sampling a roll trajectory between -40° and 40° and applying it by warping the video. We name the resulting dataset *SpatialVID-extreme*. By having a diverse set of pitches and rolls, we ensure a diverse set of text prompts and reference videos for FID and FVD computations. Please refer to the supplementary material for an evaluation of the diversity of our benchmark dataset.

For the camera trajectories fed to the models, we annotate translations using ViPE [16] and normalize them to have a unit-length maximum translation magnitude (since some methods are not trained on metric-scale data), but discard the rotation. We instead generate a random rotation by sampling roll, pitch, and yaw trajectories (see the supp. for details). This strategy allows for a broader range of rotations and also better tests the methods’ adherence to the camera trajectory. Indeed, we receive a normal environment description as input prompt,

Table 1: Quantitative results on our SpatialVID-extreme evaluation dataset. Our method, applied on different backbones and camera encodings, outperforms all baselines on absolute camera orientation metrics (PitchErr and GravityErr), and nearly all dynamic video generation methods on relative rotation error (RotErr), and is competitive on aesthetic quality (FID, FVD). Furthermore, removing the null-pitch conditioning drastically degrades absolute orientation metrics. PreciseCam + GEN3C is grayed out to emphasize it cannot generate dynamic content.

Method	Dynamic	PitchErr $\downarrow$	GravityErr $\downarrow$	RotErr $\downarrow$	TransErr $\downarrow$	CLIP $\uparrow$	FID $\downarrow$	FVD $\downarrow$
<i>Cosmos 7B + projected point cloud encoder + full finetuning</i>								
PreciseCam [5] + GEN3C [32]	×	26.65	30.80	10.99	0.40	20.2	101.2	844.1
<i>CogVideoX-2B + Plücker camera encoder + ControlNet</i>								
AC3D [2] + cam. text.	✓	39.09	44.13	29.27	0.62	23.1	113.0	984.1
AC3D [2] + cam. text. + abs. Plücker	✓	38.49	42.93	34.85	0.64	23.3	117.6	1022.6
Ours	✓	21.19	24.84	18.48	0.68	21.1	110.9	915.7
<i>WAN 2.1 1.3B + Plücker camera encoder + full finetuning</i>								
PreciseCam [5] + WAN-I2V-CC [37]	✓	29.16	33.48	24.54	0.62	22.3	109.2	776.3
Ours	✓	10.50	13.05	25.84	0.68	20.5	114.7	1041.6
<i>WAN 2.2 5B + Plücker camera encoder + full finetuning</i>								
Ours w/o null-pitch cond.	✓	14.39	16.36	17.26	0.74	22.0	110.6	895.2
Ours	✓	8.36	10.54	15.05	0.74	20.6	116.0	1014.4
<i>WAN 2.1 1.3B + UCPE encoding + UCPE module training</i>								
UCPE [53]	✓	16.25	18.88	18.38	0.44	22.5	110.7	957.9
Ours w/o null-pitch cond.	✓	18.92	21.96	11.21	0.47	22.6	110.6	979.2
Ours w/o absolute rotations	✓	50.62	56.18	14.35	0.53	21.7	108.9	862.4
Ours	✓	13.98	17.16	12.12	0.46	21.2	112.5	978.1

but for extreme camera angles, such as looking straight up, adhering to the camera conditioning likely requires generating views of the sky.

4.4 Metrics

To evaluate camera rotation controls, we first measure absolute orientation accuracy using two metrics: angular pitch error (“PitchErr”), and gravity error (“GravityErr”), defined as the angle between the ground-truth up vector and that of the generated video, in camera space. The pitch angle and up vector are obtained from each frame using Perspective Fields [18]. Roll error is not directly computed, as estimating it near the poles (looking straight up or down) is unreliable; however, roll deviations are implicitly reflected in the gravity error.

To evaluate the relative rotation error (“RotErr”), we run VGGT [39] on the generated frames and compute the average relative angular error. We reuse the same definition of rotation error as [12]. To evaluate camera motion accuracy, we compute a translation error (“TransErr”) by comparing the estimated and ground-truth trajectories; both trajectories are normalized prior to computing the error to account for scale differences. The CLIP score [14] is computed to measure the alignment between the generated video frames and the reference text description. Finally, we employ the Fréchet Inception Distance (FID) and Fréchet Video Distance (FVD) to assess the visual quality and realism of the generated frames and sequences, respectively.

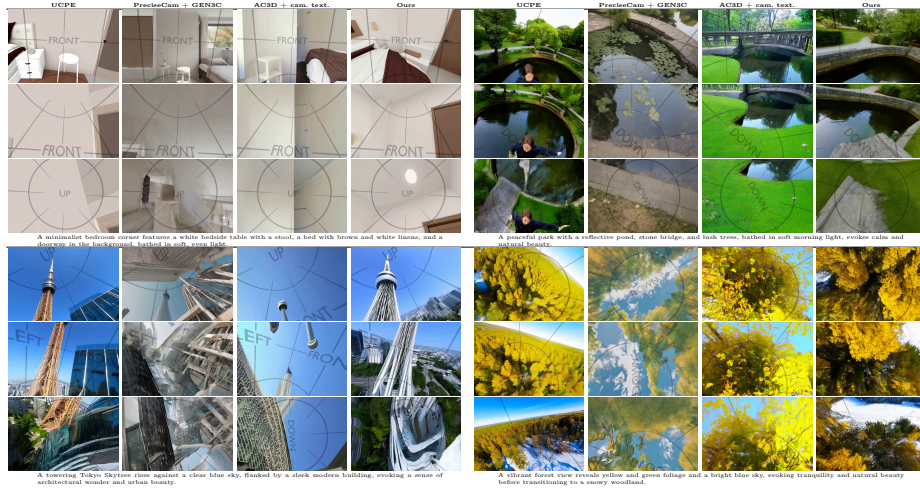


Fig. 5: Qualitative results on the SpatialVID-extreme benchmark. The input absolute camera angle is shown as a dark overlay on the image. We observe that our method aligns well with the overlay, showing the sky when the zenith (“UP”) is in the view and showing the horizon aligned with the equator line (going through “FRONT”), for example. Please refer to the supp. for additional results.

4.5 Results

We present the quantitative results of our evaluation in Tab. 1. Overall, our method outperforms all dynamic video generation baselines on absolute camera orientation metrics (“PitchErr”, “GravityErr”) and nearly all on relative rotation (“RotErr”). Since GEN3C can only synthesize novel viewpoints without scene motion, we include its results here for reference only. Our method achieves FID scores similar to those of other baselines and FVD scores on par with UCPE and AC3D. Baselines using PreciseCam as a starting point improve the FVD score, at the cost of much worse camera orientation metrics. We observe that our method results in a slight drop in CLIP score, see the discussion on prompt-camera entanglement in Sec. 4.6 below.

Representative qualitative results are illustrated in Fig. 5 (see the supplementary for more). We observe that all baselines struggle with extreme angles (e.g., the “down” starting point in the bottom-right example). Here, they attempt to generate visual content in accordance with the prompt, but this contradicts the desired camera orientation. Another example is shown in the top-left example, where the ceiling is only correctly generated with our method when the camera is pointed up, whereas UCPE, PreciseCam+GEN3C and AC3D+cam. text. generate the wall.

Ablations. We trained separate GIMBALDIFFUSION checkpoints in which we ablated null-pitch conditioning (i.e., captioning is done on the training video itself) and absolute camera conditioning (i.e., making the camera poses relative

to the first frame). We report the results in Tab. 1, in the “w/o null-pitch cond.” and “w/o absolute rotations” rows, respectively. Making cameras relative to the first frame leads to a $3.62\times$ and $3.27\times$ increase in pitch and gravity error respectively. This high error is due to the fact that the absolute extreme pitch and roll information are simply not provided to the model. Ablating the null-pitch conditioning leads to a 35% and 72% increase in pitch error, for the “WAN 2.1 1.3B + UCPE encoding” and “WAN 2.2 5B + Plücker camera encoder” variants, respectively. A similar effect is seen in the gravity error. We further investigate the effect of null-pitch conditioning in the next section.

4.6 Prompt-camera entanglement evaluation

To investigate the effect of null-pitch conditioning, we develop a second benchmark dataset to evaluate whether methods can disambiguate between two possibly conflicting inputs: the prompt and the camera pitch angle. Indeed, pitch is highly correlated with semantic concepts, e.g., a $+90^\circ$ pitch (looking up) is strongly associated with images of the sky/ceiling, whereas a -90° pitch is strongly associated with ground features. Here, roll and yaw angles are ignored, since both are uncorrelated to object semantics in the field of view.

We start by randomly selecting 20 panoramas from the PolyHaven dataset [31]. For each, we take a forward-looking crop with a 90° FoV and caption it using InternVL-3-8B [56]. We then give this prompt to each evaluated method along with a static camera orientation. The methods are evaluated on pitch angles going from -90° to $+90^\circ$, in 10° increments. In total, this benchmark consists of 380 pairs of input prompts and pitch angles.

The results obtained with baselines providing gravity-aligned camera control (UCPE [53] and PreciseCam [5]) are shown in Fig. 6. Qualitatively, we observe in Fig. 6-(a) that our proposed null-pitch conditioning enforces strict adherence to the camera orientation when the camera is pointing straight up or straight down. Quantitatively, we observe in Fig. 6-(b) (top) a substantially lower pitch-angle error across all 20 scenes, particularly at extreme absolute pitch values.

One simple way to assess the visual alignment between the text prompt and the output video is to compute the CLIP score directly between them. However, this naive approach has a key limitation: the prompt features may rightly be absent from the video if they violate the user-specified camera constraint (e.g., textual descriptions of ground content should not be visible if the camera is pointing up)—but not doing so would negatively affect the CLIP score. To address this contradiction, we compute the CLIP score between the generated video frames and a caption of the ground-truth frame (obtained with InternVL-3-8B) for each pitch angle separately, and report results in Fig. 6-(b) (bottom). We observe that null-pitch conditioning *improves* CLIP similarity at extreme pitch values, aligning results closer to the ground-truth curve. See the supplementary material for a detailed analysis.

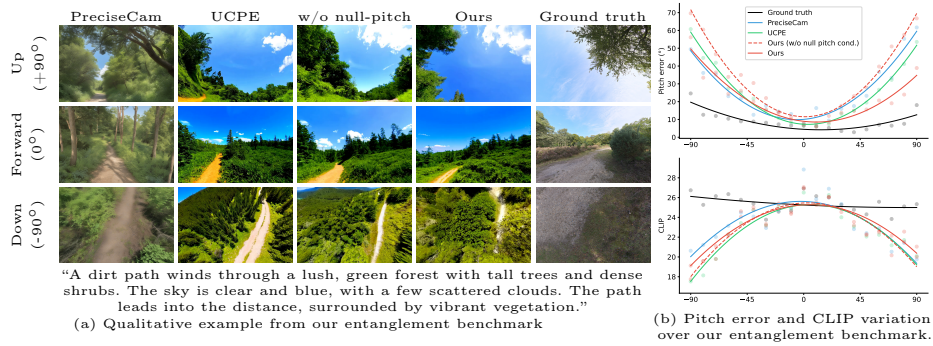


Fig. 6: Evaluating prompt-camera pitch entanglement. (a) Without careful captioning, generative models may ignore camera conditioning when it conflicts with prompt semantics, especially at extreme pitch values (e.g., -90° and $+90^\circ$). (b) Null-pitch conditioning reduces pitch error at extreme angles (top) and improves CLIP similarity to the ground truth (bottom). See the supp. for additional results.

5 Discussion

We present GIMBALDIFFUSION, a method that enables *extreme and arbitrary camera control* in text-to-video generation by grounding camera poses in a gravity-aligned absolute coordinate system. Our pipeline combines 360° panoramic videos with camera pose estimation to produce training data covering the full sphere of possible viewpoints, including extreme pitch, large roll, and full 360° rotation trajectories that were previously beyond the reach of generative video models. We also proposed null-pitch conditioning, a data generation step that enhances camera conditioning fidelity when the text prompts contain conflicting cues.

While it introduces novel capabilities for text-to-video models, our method still has a few limitations. First, our current on-the-fly camera sampling is restricted to rotations and must use the translations from the original source videos. A promising area for future research is to extend this data augmentation to include translations, potentially by leveraging recent successes in real-time novel view synthesis methods such as 3DGS [20]. Second, the generated videos can exhibit visual artifacts (for example, the distortions in the Tokyo Skytree in Fig. 5). We expect that continued progress in video generation models will mitigate these issues while remaining compatible with our framework. We hope our work lays the foundation for robust camera control under extreme rotations in video generation, empowering creators to produce dynamic storyboards with arbitrary camera orientations and convey their visual ideas with greater precision and speed.

Acknowledgements. This research was supported by Adobe and a Natural Sciences and Engineering Research Council of Canada (NSERC) scholarship, ref. 600578. Computing resources were provided by Adobe and the Digital Research Alliance of Canada. We thank Yohan Poirier-Ginter, Qitao Zhao, Rahul Sajjani, Jack Hilliard and Jonathan Roussel for discussions and proofreading.

References

1. Acharya, D., Huang, Z., Paudel, D.P., Gool, L.V.: Towards high resolution video generation with progressive growing of sliced wasserstein gans. In: CoRR. vol. abs/1810.02419 (2018)
2. Bahmani, S., Skorokhodov, I., Qian, G., Siarohin, A., Menapace, W., Tagliasacchi, A., Lindell, D.B., Tulyakov, S.: Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2025)
3. Bahmani, S., Skorokhodov, I., Siarohin, A., Menapace, W., Qian, G., Vasilkovsky, M., Lee, H.Y., Wang, C., Zou, J., Tagliasacchi, A., Lindell, D.B., Tulyakov, S.: VD3D: Taming large video diffusion transformers for 3d camera control. In: Int. Conf. Learn. Represent. (2025)
4. Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X., Wan, P., et al.: Recammaster: Camera-controlled generative rendering from a single video. In: IEEE/CVF Int. Conf. Comput. Vis. (2025)
5. Bernal-Berdun, E., Serrano, A., Masia, B., Gadelha, M., Hold-Geoffroy, Y., Sun, X., Gutierrez, D.: PreciseCam: Precise camera control for text-to-image generation. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2025)
6. Chang, D., Shi, Y., Gao, Q., Fu, J., Xu, H., Song, G., Yan, Q., Zhu, Y., Yang, X., Soleymani, M.: MagicPose: Realistic Human Poses and Facial Expressions Retargeting with Identity-aware Diffusion. In: Int. Conf. Mach. Learn. (2024)
7. Chen, W., Wu, J., Xie, P., Wu, H., Li, J., Xia, X., Xiao, X., Lin, L.: Control-a-video: Controllable text-to-video generation with diffusion models. In: CoRR (2023)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2017)
9. Do, T., Vuong, K., Park, H.S.: Egocentric scene understanding via multimodal spatial rectifier. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2022)
10. Fortier-Chouinard, F., Zhang, Z., Messier, L.E., Garon, M., Bhattad, A., Lalonde, J.F.: Spotlight: Shadow-guided object relighting via diffusion. In: Int. Conf. 3D Vis. (2026)
11. Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., Wang, W., Liu, Y.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In: ACM SIGGRAPH Conf. (2025)
12. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: CameraCtrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2024)
13. He, H., Yang, C., Lin, S., Xu, Y., Wei, M., Gui, L., Zhao, Q., Wetzstein, G., Jiang, L., Li, H.: CameraCtrl ii: Dynamic scene exploration via camera-controlled video diffusion models. arXiv preprint arXiv:2503.10592 (2025)
14. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: Conf. Emp. Metho. Nat. Lang. Proc. (2021)
15. Hu, L., Gao, X., Zhang, P., Sun, K., Zhang, B., Bo, L.: Animate anyone: Consistent and controllable image-to-video synthesis for character animation. arXiv preprint arXiv:2311.17117 (2023)
16. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., Ren, J., Xie, K., Biswas, J., Leal-Taixe, L., Fidler, S.: Vipe: Video pose engine for 3d geometric perception. In: NVIDIA Research Whitepapers (2025)

17. Jiang, F., et al.: Megasam: Scaling up camera pose estimation with a foundation model for structure-from-motion. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2025)
18. Jin, L., Zhang, J., Hold-Geoffroy, Y., Wang, O., Blackburn-Matzen, K., Sticha, M., Fouhey, D.F.: Perspective fields for single image camera calibration. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2023)
19. Kant, Y., Siarohin, A., Wu, Z., Vasilkovsky, M., Qian, G., Ren, J., Guler, R.A., Ghanem, B., Tulyakov, S., Gilitschenski, I.: Spad: Spatially aware multi-view diffusers. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2024)
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4), 139–1 (2023)
21. Kluger, F., Ackermann, H., Yang, M.Y., Rosenhahn, B.: Temporally consistent horizon lines. In: *Int. Conf. Robot. Autom.* (2020)
22. Kocsis, P., Philip, J., Sunkavalli, K., Nießner, M., Hold-Geoffroy, Y.: LightIt: Illumination modeling and control for diffusion models. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2024)
23. Kong, X., Liu, S., Lyu, X., Taher, M., Qi, X., Davison, A.J.: Eschnet: A generative model for scalable view synthesis. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2024)
24. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with Mast3r. In: *Eur. Conf. Comput. Vis.* (2024)
25. Li, R., Yi, B., Liu, J., Gao, H., Ma, Y., Kanazawa, A.: Cameras as relative positional encoding. In: *Adv. Neural Inform. Process. Syst.* (2025)
26. Li, Y., Min, M., Shen, D., Carlson, D., Carin, L.: Video generation from text. In: *Assoc. Adv. of Art. Int.* (2018)
27. Liao, K., Wu, S., Wu, Z., Jin, L., Wang, C., Wang, Y., Wang, F., Li, W., Loy, C.C.: Thinking with camera: A unified multimodal model for camera-centric understanding and generation. In: *Int. Conf. Learn. Represent.* (2026)
28. Magar, N., Hertz, A., Tabellion, E., Pritch, Y., Rav-Acha, A., Shamir, A., Hoshen, Y.: LightLab: Controlling light sources in images with diffusion models. In: *ACM SIGGRAPH Conf.* (2025)
29. Miyato, T., Jaeger, B., Welling, M., Geiger, A.: Gta: A geometry-aware attention mechanism for multi-view transformers. In: *Int. Conf. Learn. Represent.* (2024)
30. Moulon, P., Monasse, P., Perrot, R., Marlet, R.: Openmvg: Open multiple view geometry. In: *Int. Work. Reproduc. Res. Patt. Recog.* (2016)
31. Poly Haven: Poly Haven HDRIs: Free high-dynamic-range environment maps. <https://polyhaven.com/hdri/> (2025), accessed: 2025-11-12
32. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2025)
33. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2016)
34. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* (2024)
35. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: Decomposing motion and content for video generation. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2018)
36. Veicht, A., Sarlin, P.E., Lindenberger, P., Pollefeys, M.: GeoCalib: Single-image calibration with geometric optimization. In: *Eur. Conf. Comput. Vis.* (2024)

37. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: WAN: Open and Advanced Large-Scale Video Generative Models. arXiv preprint arXiv:2503.20314 (2025)
38. Wang, J., Yuan, Y., Zheng, R., Lin, Y., Gao, J., Chen, L.Z., Bao, Y., Zhang, Y., Zeng, C., Zhou, Y., Long, X., Zhu, H., Zhang, Z., Cao, X., Yao, Y.: SpatialVID: A large-scale video dataset with spatial annotations. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2026)
39. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2025)
40. Wang, Q., Li, W., Mou, C., Cheng, X., Zhang, J.: 360dvd: Controllable panorama video generation with 360-degree video diffusion model. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2024)
41. Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2025)
42. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2024)
43. Weng, L.: Diffusion models for video generation. Lil'Log (blog) (2024), <https://lilianweng.github.io/posts/2024-04-12-diffusion-video/>
44. Workman, S., Zhai, M., Jacobs, N.: Horizon lines in the wild. In: Brit. Mach. Vis. Conf. (2016)
45. Wu, C., et al.: Visualsfm: A visual structure from motion system (2011)
46. Xia, Y., Weng, S., Yang, S., Liu, J., Zhu, C., Teng, M., Jia, Z., Jiang, H., Shi, B.: Panowan: Lifting diffusion video generation models to 360° with latitude/longitude-aware mechanisms. In: Adv. Neural Inform. Process. Syst. (2025)
47. Xian, W., Li, Z., Fisher, M., Eisenmann, J., Shechtman, E., Snavely, N.: Uprightnet: geometry-aware camera orientation estimation from single images. In: IEEE/CVF Int. Conf. Comput. Vis. (2019)
48. Xing, J., Mai, L., Ham, C., Huang, J., Mahapatra, A., Fu, C.W., Wong, T.T., Liu, F.: Motioncanvas: Cinematic shot design with controllable image-to-video generation. In: ACM SIGGRAPH Conf. (2025)
49. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)
50. Yenphraphai, J., Pan, X., Liu, S., Panozzo, D., Xie, S.: Image sculpting: Precise object editing with 3D geometry control. In: IEEE/CVF Conf. Comput. Vis. Pattern Recog. (2024)
51. YU, M., Hu, W., Xing, J., Shan, Y.: TrajectoryCrafter: Redirecting camera trajectory for monocular videos via diffusion models. In: IEEE/CVF Int. Conf. Comput. Vis. (2025)
52. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. IEEE Trans. Pattern Anal. Mach. Intell. (2025)

53. Zhang, C., Li, B., Wei, M., Cao, Y.P., Gambardella, C.C., Phung, D., Cai, J.: Unified camera positional encoding for controlled video generation (2025), <https://arxiv.org/abs/2512.07237>
54. Zhang, C., Liang, H., Chen, D.Y., Wu, Q., Plataniotis, K.N., Gambardella, C.C., Cai, J.: Panflow: Decoupled motion control for panoramic video generation. In: Assoc. Adv. of Art. Int. (2026)
55. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. In: ACM SIGGRAPH Conf. (2018)
56. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025)