Can Generative Video Models Help Pose Estimation?

Ruojin Cai[1,2] Jason Y. Zhang[1] Philipp Henzler[1] Zhengqi Li[1]
Noah Snavely[1,2] Ricardo Martin-Brualla[1]

[1]Google [2]Cornell University

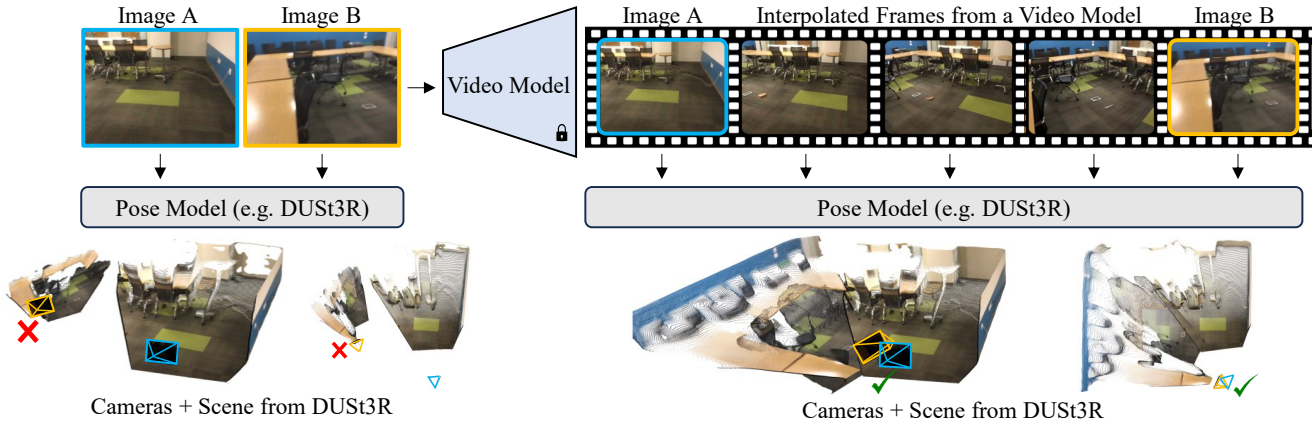


Figure 1. **Improving pose estimation by interpolating frames using a video model.** Given two images of a scene with almost no overlap, we aim to recover their relative camera pose. Without being able to rely on visual correspondences, existing methods struggle in this setting (left). We propose to use an off-the-shelf video generation model to interpolate a video connecting the two images. Augmented with the frames generated by the video model, existing pose estimators (e.g. DUSt3R [60]) are able to more accurately recover the correct pose (right).

Abstract

Pairwise pose estimation from images with little or no overlap is an open challenge in computer vision. Existing methods, even those trained on large-scale datasets, struggle in these scenarios due to the lack of identifiable correspondences or visual overlap. Inspired by the human ability to infer spatial relationships from diverse scenes, we propose a novel approach, InterPose, that leverages the rich priors encoded within pre-trained generative video models. We propose to use a video model to hallucinate intermediate frames between two input images, effectively creating a dense, visual transition, which significantly simplifies the problem of pose estimation. Since current video models can still produce implausible motion or inconsistent geometry, we introduce a self-consistency score that evaluates the consistency of pose predictions from sampled videos. We demonstrate that our approach generalizes among three state-of-the-art video models and show consistent improvements over the state-of-the-art DUSt3R baseline on four diverse datasets encompassing indoor, outdoor, and object-centric scenes. Our findings suggest a promising avenue for improving pose estimation models by leveraging large generative models trained on vast amounts of video data, which is more readily available than 3D data. See our project page for results: `Inter-Pose.github.io`.

1. Introduction

Consider the classroom in Fig. 1. We, as humans, can reasonably guess the spatial relationship between the two images, recognizing that the table on the left side of the first image is the same as the table on the right side of the second image. Even though the images are taken from viewpoints with almost no overlap, we leverage our prior knowledge about typical classroom layouts to infer this connection. This task of determining the relative pose between two images is a core component of all pose estimation pipelines and a pre-requisite for most tasks in 3D computer vision.

Traditional approaches to pairwise pose estimation rely on identifying and matching features between an image pair [35] to compute the relative geometric transformation [18]. While effective for images with significant overlap and texture, these methods struggle when faced with drastically different viewpoints, as seen in our classroom

example. Recent advances in deep learning have led to more robust pose estimators. The groundbreaking DUSt3R [60] model is trained on a mixture of several large-scale 3D datasets and demonstrates impressive performance and generalization ability. However, even such a sophisticated method struggles with extreme viewpoint changes where establishing correspondences becomes impossible.

Unlike 3D understanding models like DUSt3R, video models can be pre-trained on vast amounts of web-scale video data, orders of magnitude larger than 3D datasets. The scale of the data allows for training models that learn significantly more powerful priors of the visual world compared to 3D understanding models. For instance, state-of-the-art video models can generate videos with complex camera motions moving through a scene, reflections on shiny materials, and dynamic subjects undergoing complex interactions, and they can be prompted by images or text. Our goal is to tap into this extracted knowledge for downstream scene understanding tasks, like pose estimation.

An exciting application of such generative video models is to generate videos that interpolate between two given key frames. Thanks to the learned visual prior, the generated interpolated videos can display plausible, 3D consistent camera motions that transform one video into another. We observe that such hallucinations provide an explanation of the the scene. In this paper, we propose InterPose, which demonstrates that feeding generated interpolated frames along with the original input pair to state-of-the art pose estimation methods can improve their robustness and accuracy over using the original pair alone.

In some cases, generated videos contain visual inconsistencies, like morphing or shot cuts, that can degrade pose estimation performance. One approach is to sample multiple video interpolations, with the hope that one displays a plausible interpretation of the scene that is 3D consistent. However, how do we tell whether a video sample is good?

We address this by introducing a self-consistency score to evaluate the reliability of the predicted pose for a given video. Our method samples different sets of frame indices from the interpolated video, and computes multiple pose estimates using these frames together with the input image pair, creating multiple pose estimates per sampled video. An ideal pose prediction comes from a video whose pose estimates are invariant to the specific sampled frame indices, e.g., whose pose estimates are tightly clustered, and among the pose estimates from that video, one that is close to the other estimates, e.g., the centroid or medoid.

Although simple, we demonstrate the efficacy of our method on challenging input pairs extracted from four diverse datasets, including indoor, outdoor and object-centric scenes. In summary, our key contributions include:

- we demonstrate for the first time that a generative video model can improve pose estimation by acting as a world prior, improving on the results of a state-of-the-art pose estimator (DUSt3R);
- we present a new benchmark of challenging image pairs with small to no overlap across four different datasets encompassing outdoor scenes, indoor scenes, and object-centric views;
- and we propose a simple-yet-effective way to score the self-consistency of estimated poses from interpolated videos that generalizes across three different publicly available video models.

2. Related work

2.1. Generative Video Models

Early efforts to build video generators based on GANs [29, 44, 54, 58] and VAEs [13, 22, 56] had limited visual fidelity. More recently, diffusion models [19, 49, 50] have revolutionized generative image [39, 40, 43] and video generation. Earlier diffusion-based models often made predictions directly in pixel space [20, 21, 48]. Such architectures made it computationally expensive to predict high resolution image frames. To alleviate this issue, subsequent works looked at making predictions in the latent space of an autoencoder [3, 7, 17, 57, 62]. Since then a variety of video models has been released that demonstrates near-photorealism at high resolution. These models are only available behind a paywall [28, 36, 42] or are not available to the public at all [9]. In our work, we evaluate both public and commercial video models.

2.2. Relative Pose Estimation

The classic approach to computing the pose between two images is to extract image features [5, 35, 41], find correspondences [37], and then compute the fundamental matrix [18, 34, 38] while rejecting outliers [16]. Learning-based methods have significantly improved each of these components, providing better features [14, 55] and matchers [24, 26, 32, 45] or even learning the correspondences directly [51–53]. While these bottom-up approaches are capable of achieving pixel-perfect alignment, their reliance on correspondences make them brittle and require salient visual overlap between the images.

With the advent of deep learning, top-down pose estimation models trained on large-scale 3D datasets can learn to estimate relative pose between images with wide baselines [6, 10]. A key challenge is that the relative pose is often ambiguous. Recent works have explored handling pose estimation probabilistically using factorized distributions [11], energy-based models [31, 64], or diffusion [59, 65]. More recent approaches have transitioned to distributed ray- or point-based representations of pose to great effect [4, 30, 60, 65, 66]. Because these methods rely on 3D datasets with limited diversity, finding data for

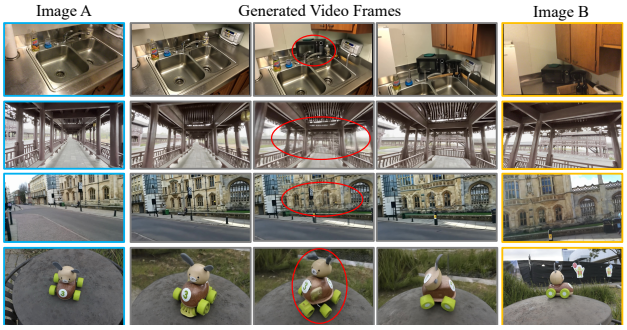


Figure 2. **Common failure modes of video models.** We show some failure modes of interpolating between two images. In the first row, a microwave suddenly appears over the sink. In the second and third row, the video model morphs and blends images without consistent changes to the underlying scene geometry. In the fourth row, the object's appearance changes in an unrealistic way.

generalization across all scene distributions is an open challenge. DUSt3R [60] leverages CroCo pre-training [61] and a transformer architecture to predict per-image point maps relative to the camera coordinate frame of the first image. Subsequently, camera poses can be recovered from these predicted point maps. MicKey [4] and MASt3R [30] further improve pose estimation by incorporating local feature extraction and enhancing feature matching. We view these methods as complementary to our work and in fact, we make direct use of DUSt3R [60] and MASt3R [30] as video models can bridge the distribution gap but cannot recover poses by themselves.

3. Method

Given two images I_A and I_B, our goal is to recover their relative camera pose. We introduce InterPose, which leverages off-the-shelf video models to generate the intermediate frames between the two images. By using these generated frames alongside the original image pair as input to a camera pose estimator, we provide additional context that can improve pose estimation compared to just using the two input images. A key challenge is that the generated videos may contain visual artifacts or implausible motion. Thus, we generate multiple videos which we score using a self-consistency metric to select the best video sample.

3.1. Preliminaries

Pose parameterization. Given two images I_A and I_B associated with ground truth world-to-camera transformations T_A and T_B:

$$T_A = \begin{bmatrix} R_A & t_A \\ 0 & 1 \end{bmatrix}, \quad T_B = \begin{bmatrix} R_B & t_B \\ 0 & 1 \end{bmatrix}, \tag{1}$$

we aim to recover their relative pose $T_{\text{rel}} = T_B T_A^{-1}$, where the relative rotation and translation are $R_{\text{rel}} = R_B R_A^{-1}$ and $t_{\text{rel}} = t_B - R_{\text{rel}} t_A$, respectively.

The distance between two pose transforms T_1 and T_2 can be computed by summing their geodesic rotation and translation angle error. Note that translation angle error makes the distance invariant to scale, and is typically used for pose evaluation.

$$\text{dist}(T_1, T_2) = \text{dist}_R(R_1, R_2) + \text{dist}_t(R_1, R_2), \tag{2}$$

$$\text{dist}_R(R_1, R_2) = \arccos\left(\frac{\text{Trace}(R_2 R_1^\top) - 1}{2}\right), \tag{3}$$

$$\text{dist}_t(t_1, t_2) = \arccos\left(\left|\frac{t_1}{\|t_1\|} \cdot \frac{t_2}{\|t_2\|}\right|\right). \tag{4}$$

Camera pose estimator. In the following, we assume a black-box camera pose estimator, that given N images returns estimated relative poses across all N images. In practice, we use DUSt3R [60] and MASt3R [30], but other options could be possible, including non-learning based ones like COLMAP [46, 47]. While the core algorithms of DUSt3R and MASt3R are designed for pairwise image matching, we utilize their multi-view extensions, which perform post-processing optimization over point clouds and poses to estimate poses for an entire image set. Henceforth, we refer to these multiview extensions as DUSt3R and MASt3R. We denote the pose estimators as:

$$f_{\text{pose}}(\{I_A, I_B, I_1, \ldots, I_{N-2}\}) = \hat{T}_{\text{B}} \hat{T}_{\text{A}}^{-1} = \hat{T} \tag{5}$$

that takes the input pair I_A, I_B, with optionally additional frames I_i, and outputs the relative pose from I_A to I_B.

Generative video models. We use a generative video model f_{vid} capable of interpolating between image frames:

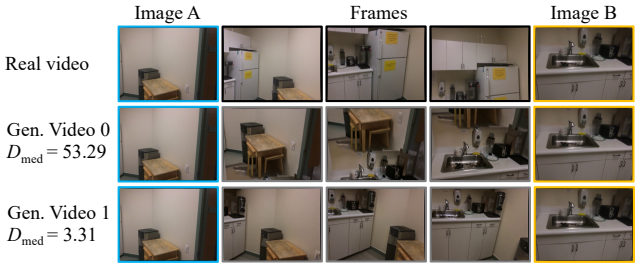
$$f_{\text{vid}}(I_A, I_B, p) = [I_1, I_2, \ldots, I_N] \tag{6}$$

where I_1=I_A, I_N=I_B, and p is a text prompt. We consider 3 video models: DynamiCrafter [62], Runway Gen-3 Alpha Turbo [42], and Luma Dream Machine [36]. We generate multiple samples per input pair (I_A, I_B) by providing different prompts or orderings of the input pair.

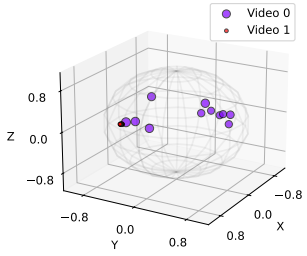
3.2. Self-consistency Score

Video models generate wildly varying results for similar inputs. This variability is particularly present when doing video interpolation, where a number of camera paths and scene configurations are possible, especially in the low- or no overlap case. Furthermore, the quality of the different samples varies a lot, and artifacts and inconsistencies (e.g., objects appearing/disappearing) are common, as shown in Fig. 2. To address these issues, we propose a two-pronged approach: 1) we generate n different videos to account for inherent variability, and 2) we develop a score to identify the video that exhibits the most consistent structure.

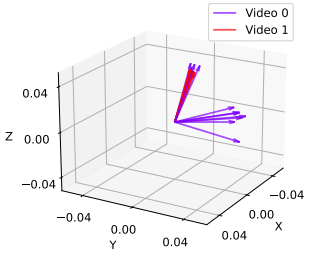
Determining consistent videos. Consider a low-quality video that has rapid shot-cuts or inconsistent geometry



(a) We take images A and B and generate interpolated videos, (two, Video 0 and Video 1, are shown here for illustration). In this case, the ground truth real video is available, and so we show it at the top for comparison.



(b) Visualization of predicted rotations using randomly sampled subsets of each generated video on the unit sphere. Note that the samples from Video 1 cluster tightly, and so appear as nearly a single point.



(c) Visualization of predicted translation directions using randomly sampled subsets of frames from Video 0 and Video 1.

Figure 3. **Self-consistency scores for poses derived from generated videos.** (a) From a pair of input frames A and B, we generate several candidate videos from a given video interpolation method. For each video, we sample subsets of frames and compute a relative pose from A to B from each subset ((b) and (c)). We then compute a medoid distance between these samples as a *self-consistency score* for that video, shown to the left of each video in part (a). In this case, Video 0 contains artifacts, and so yields an inconsistent set of poses (and a high medoid distance), which Video 1 is much more natural and produces a more consistent set of poses and a lower medoid distance.

(Fig. 2). Selecting different subsets of frames from that video would likely produce dramatically different pose estimations. We operationalize this concept by measuring a video's "self-consistency."

For a given sampled video, we randomly select m sets of k frames (always including the original input images I_A and I_B), and calculate the predicted relative pose for each frame subset:

$$f_{\text{pose}}(\{I\}^{(i)}) = \hat{T}^{(i)}. \tag{7}$$

We quantify video inconsistency using the medoid distance:

$$D_{\text{med}} = \min_i \frac{1}{m-1} \sum_{j \neq i} \text{dist}\left(\hat{T}^{(i)}, \hat{T}^{(j)}\right). \tag{8}$$

Intuitively, a low medoid distance indicates that every subset of frames produces roughly the same relative pose between I_A and I_B, suggesting a consistent video. We illustrate this concept in Fig. 3.

In some degenerate cases, a video that is generated poorly (e.g. only has blurry or uninformative frames) could still have low medoid distance if it consistently makes blatantly incorrect predictions (e.g., always 180 degrees apart). Thus, we found it helpful to bias the metric so that the medoid should not deviate too far from the pose estimated from the original input images alone:

$$D_{\text{total}} = D_{\text{med}} + \text{dist}\left(\hat{T}_{\text{med}}, f_{\text{pose}}(\{I_A, I_B\})\right), \tag{9}$$

where $\hat{T}_{\text{med}}$ is the medoid relative pose.

Putting it all together. We select the video with the lowest D_{total}, and output as the consensus pose the predicted medoid relative pose $\hat{T}_{\text{med}}$.

3.3. Method Overview and Implementation Details

Given a pair of images, we first generate n videos using a video generative model. For each generated video, we sample subsets of k frames (2 original input images and $k-2$ generated frames) to compute candidate poses using a pose estimator (e.g., DUSt3R or MASt3R). This process is repeated m times, yielding m candidate poses per video. Finally, we select the most reliable prediction based on the medoid distance metric among all candidates.

For each image pair I_A and I_B, we use GPT-4o [1] to generate two different captions to describe the content of the input image ("Use one sentence to caption these images of the same static scene" and "Use simple language to specifically include details that describe the same scene shown in these two images in one sentence"). We then use the captions to generate interpolated videos for both the original (I_A to I_B) and the flipped order (I_B to I_A). We found this flipping to be crucial because video models are often biased toward producing videos that pan to the right as opposed to the left (see Fig. 6).

These generated video prompts guide the video models to produce coherent intermediate frames (see Fig. 4). Using each of the four generated prompts, we run each video model to interpolate in the specified direction, resulting in a total of $n = 4$ generated videos per image pair. For each generated video, we sample subsets of $k = 5$ images (2 original input, 3 generated) to compute candidate poses. In particular, we sample subsets of frames randomly 10 times and once with uniform spacing, for a total of $m = 11$ sampled frame subsets per video. For each sample, the $k = 5$ frames are provided as input to DUSt3R, and from the resulting poses we compute the medoid as described above.

4. Experiments

4.1. Dataset and Benchmark

We evaluate our method, InterPose, on challenging inputs from four datasets annotated with ground truth 3D camera poses, covering a diverse range of indoor and outdoor setups. For each dataset, we selected image pairs by randomly sampling frames within a specified delta yaw range

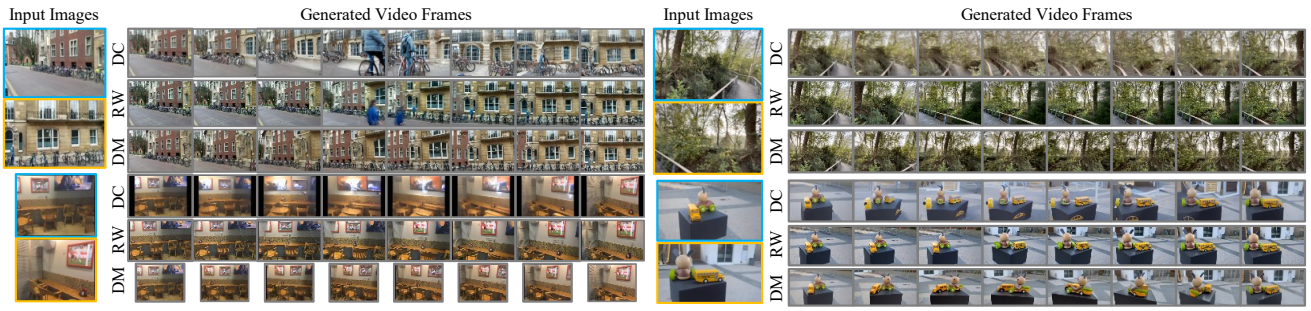


Figure 4. Qualitative comparison of the three video models: DynamiCrafter (DC), Runway (RW), and Dream Machine (DM), using the same text prompt for each video model. Top left: a pair of images from the Cambridge Landmarks dataset. Prompt: Dozens of bicycles are parked along the street in front of old brick and stone buildings, with a person walking by and trees in the background. Bottom left: a pair of images from ScanNet. Prompt: A cozy café corner features wooden chairs, framed sports photos, and a TV screen. Top right: a pair from DL3DV-10K. Prompt: A peaceful morning stroll along a wooden boardwalk surrounded by lush, sunlit greenery. Bottom right: a pair from NAVI. Prompt: A wooden toy figure with gray ears and green wheels sits next to a small yellow school bus on a black pedestal in an outdoor paved area.

(see below). This selection ensures challenging pose estimation scenarios with sufficiently large viewpoint changes. Due to the prohibitive cost of running commercial video models, we limit the evaluation to at most 300 image pairs per dataset. We will release the selected indices for reproducibility.

Cambridge Landmarks [27]: This outdoor, scene-scale video dataset captures streets and building facades in Cambridge. We utilize a subset of 290 image pairs from [6] with yaw changes between 50° and 65°. These pairs feature small to no overlap, with motions characterized predominantly by rotation but minimal camera translation. Thus, we report only rotation metrics for this dataset.

ScanNet [12]: An indoor, scene-scale video dataset capturing various indoor environments. We randomly selected 300 image pairs from test 75 scenes, with yaw changes in the range of 50° and 65°.

DL3DV-10K [33]: A scene-scale, center-facing video dataset comprising over 10K videos from 65 types of point-of-interest locations. We randomly selected 300 pairs from 300 outdoor scenes, each with yaw changes ranging from 50° to 90°.

NAVI [23]: An object-centric, center-facing dataset that includes video and multiview images captured using various camera devices under different environmental conditions. We randomly selected 300 pairs from 36 objects, each with yaw changes between 50° and 90°.

While all datasets feature significant viewpoint changes, the center-facing nature of DL3DV-10K and NAVI leads to large overlaps in the view frustrums between input views. Our experiments indicate that these center-facing datasets are significantly easier for pose prediction than ScanNet and Cambridge Landmarks, which have many outward-facing camera viewpoints.

4.2. Experimental Variants

4.2.1 Baselines and Our Method

We compare our method against several pose estimators:
SIFT [35] + Nearest Neighbors: As a classic geometric baseline, we match SIFT features using nearest neighbors and RANSAC [16] to filter outliers. Using ground truth intrinsics, we compute the essential matrix, from which we extract relative rotations and translations using OpenCV [8].

LOFTR [51]: LOFTR uses a transformer to learn semi-dense matches between images. As with the SIFT baseline, we filter outliers and use the correspondences to estimate an essential matrix.

DUSt3R [60]: DUSt3R is a recent method for pose estimation and 3D reconstruction from unconstrained image collections. From any number of input images, DUSt3R reconstructs a dense pointmap for each pair of images and then jointly optimizes the camera poses and to best align the point clouds.

MASt3R [30]: MASt3R, a recent follow-up method to DUSt3R, follows a similar backbone and training scheme as DUSt3R but incorporates additional heads to produce local features and facilitate feature matching. With these enhancements, MASt3R can be more accurate than DUSt3R, particularly when sufficient correspondences are available.

Ours (DUSt3R / MASt3R): We use the relative transformation predicted from the generated video with the lowest total medoid distance (see Sec. 3.2). We apply this to pose estimators DUSt3R and MASt3R.

4.3. Video Models

We evaluate three video models (visualized in Fig. 4):
DynamiCrafter [62]: DynamiCrafter is an open-source image animation model enabling video generation and keyframe interpolation. DynamiCrafter is based on a pre-trained text-to-video diffusion model and finetuned on We-

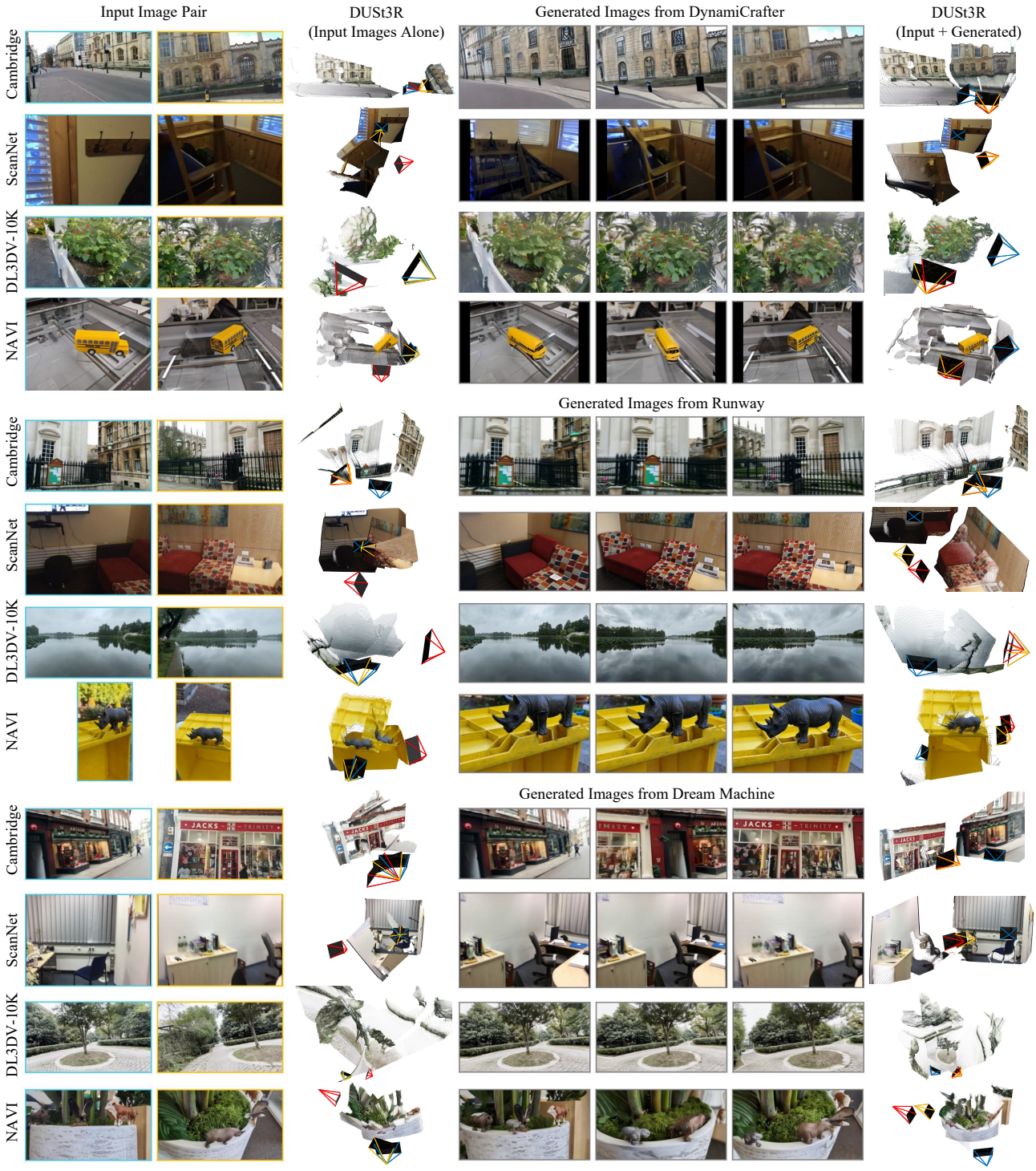


Figure 5. **Qualitative results of pose estimation from DUSt3R taking only image pair as input and taking additional video frames.** We show the input image pair in the first two columns, and the DUSt3R prediction using the image pair alone in the third column. The 3D reconstruction shows the predicted point maps and camera poses for the input images, with the first camera denoted in blue, the second camera in gold, and its corresponding ground truth camera in red, best seen digitally. In columns four to six, we visualize interpolated frames from three different video models. In the last column, we show the DUSt3R pose predictions made using all 5 images, but we are only showing the poses and pointmaps corresponding to the input images for clarity.

Pose estimator	Input data	Cambridge Landmarks					ScanNet								
		MRE↓	R_{acc} ↑			AUC_{30} ↑	MRE↓	MTE↓	R_{acc} ↑			t_{acc} ↑			AUC_{30} ↑
			5°	15°	30°				5°	15°	30°	5°	15°	30°	
SIFT+N.N.	Pair	97.64	15.17	22.41	24.48	20.49	112.95	48.99	2.06	3.44	5.50	23.02	25.09	31.62	1.82
LOFTR		30.30	31.38	56.55	70.00	51.63	64.46	45.49	8.33	17.00	22.00	27.00	28.33	35.33	6.43
DUSt3R		13.28	63.45	87.24	88.97	77.23	21.31	24.72	65.33	76.33	79.00	48.33	68.33	73.67	60.34
MASt3R		36.55	28.62	64.83	74.14	55.69	24.35	17.93	44.00	73.33	79.67	38.00	67.33	77.67	55.10
Ours (DUSt3R)	DynamiCrafter	12.70	65.17	88.97	90.34	79.00	18.96	16.42	68.00	82.33	84.33	48.67	71.67	80.33	62.14
	Runway	10.78	64.83	91.03	94.14	80.59	19.93	16.31	67.67	81.33	84.33	51.00	72.33	80.67	61.83
	Dream Machine	11.96	57.93	89.66	92.76	78.67	17.65	15.88	68.67	81.33	85.33	47.67	71.33	82.33	63.06
Ours (MASt3R)	DynamiCrafter	31.43	34.83	70.00	76.55	60.03	21.97	16.48	53.00	75.67	80.00	40.67	70.33	80.00	57.90
	Runway	29.04	42.07	72.76	78.97	63.57	21.68	15.28	50.33	75.67	81.67	41.00	70.00	83.33	57.19
	Dream Machine	27.47	34.48	74.14	80.69	63.14	19.91	15.05	53.00	78.67	83.00	41.00	70.33	82.33	58.28

Table 1. **Camera pose estimation results on outward-facing datasets (Cambridge Landmarks and ScanNet).** We evaluate the pairwise pose estimation task using our method based on two pose estimators DUSt3R and MASt3R. Our method consistently outperforms both DUSt3R and MASt3R when using input pairs alone across three video generators.

Pose estimator	Input data	DL3DV-10K								NAVI							
		MRE↓	MTE↓	R_{acc} ↑			t_{acc} ↑			$AUC_{30°}$↑	MRE↓	MTE↓	R_{acc} ↑			t_{acc} ↑			$AUC_{30°}$↑
				5°	15°	30°	5°	15°	30°				5°	15°	30°	5°	15°	30°	
SIFT+N.N.	Pair	76.64	46.80	18.06	28.09	33.44	31.77	33.11	36.45	12.11	107.46	45.10	4.67	6.67	7.33	16.33	17.00	19.00	3.20
LOFTR		35.92	41.76	37.67	52.33	61.00	40.00	41.00	45.33	23.53	71.34	51.21	6.67	14.33	19.00	24.67	25.33	29.33	4.88
DUSt3R		10.72	13.08	39.67	87.33	94.00	55.33	83.67	89.00	66.99	8.65	7.88	68.67	92.67	94.67	69.00	92.33	95.00	78.66
MASt3R		4.13	3.88	83.67	98.00	99.33	88.33	95.33	97.00	87.22	5.59	5.23	71.67	94.33	98.00	69.67	96.00	98.00	80.84
Ours (DUSt3R)	DynamiCrafter	10.02	9.13	38.33	87.33	95.67	58.33	87.00	93.00	67.97	8.26	6.57	68.00	92.67	95.67	69.00	91.67	96.67	78.78
	Runway	9.49	8.81	41.33	90.33	96.67	57.33	86.67	92.33	69.44	8.08	6.24	67.67	93.67	96.00	67.67	93.33	97.00	79.02
	Dream Machine	9.13	8.72	41.33	90.33	96.33	57.67	86.33	94.67	69.11	7.85	6.51	69.33	93.67	95.33	71.00	93.00	95.67	79.06
Ours (MASt3R)	DynamiCrafter	4.49	4.04	81.33	98.67	99.33	86.33	95.67	97.67	85.86	5.29	5.61	69.00	96.67	98.67	63.00	95.67	98.67	80.21
	Runway	4.17	4.01	81.67	99.00	99.33	87.33	96.00	97.33	86.79	5.28	5.20	72.67	96.33	98.67	69.00	97.00	98.67	81.63
	Dream Machine	4.30	4.21	80.67	99.00	99.33	85.33	94.67	97.00	85.88	5.66	5.45	70.00	97.33	98.33	70.00	96.00	98.33	81.42

Table 2. **Camera pose estimation results on center-facing datasets (DL3DV-10K and NAVI).** MASt3R demonstrates significantly improved performance on these center-facing datasets compared to outward-facing ones. We evalute our method based on two pose estimators DUSt3R and MASt3R. Our method obtains comparable results on the DL3DV-10K dataset and slightly better performance on the NAVI dataset, demonstrating that using a video model does not hinder performance even when DUSt3R and MASt3R are already strong.

bVid10M [2] for video generation from images and text prompts. Given an image pair and text prompt, DynamiCrafter generates 16 frames of resolution 320×512.

Runway [42]: Runway Gen-3 Alpha Turbo model is a commercial video generation model to generate video from text and images. The output video has 112 frames of 1280×768.

Luma Dream Machine [36]: Luma Dream Machine is a commercial video generation model that generates video from text and images. The generated video is 114 frames with the same aspect ratio as the input, and approximately one megapixel resolution.

In total, we spent $5,500 on generating prompts and running the commercial video models.

4.4. Metrics

For each pair of images, we evaluate the pose accuracy. We compute the geodesic rotation error and translation angle error using eqs. (3) and (4) respectively. We report the mean rotation error (**MRE**) and mean translation error (**MTE**) in degrees. We also evaluate the percentage of rotation ($\mathbf{R_{acc}}$) and translation ($\mathbf{t_{acc}}$) errors that are within 5°, 15°, and 30° of the ground truth. Finally, we report the Area-Under-Curve ($\mathbf{AUC}_{30}$) from 0° to 30° at 1° thresholds for rotation and translation accuracy following [25, 59].

4.5. Quantitative Results

In Table 1 and Table 2, we present a quantitative evaluation of camera pose estimation on challenging subsets of image pairs on four diverse datasets.

Baseline comparison. Feature matching-based methods like SIFT+NN and LOFTR struggle when the input pair shares little-to-no overlap as they rely on visual correspondences between overlapping regions. DUSt3R and MASt3R show significant improvements over SIFT+NN and LOFTR since they were trained on diverse 3D data without relying solely on explicit feature correspondences.

Performance with Generative Video Models. We find that our method of combining generative video models with a pose estimator consistently enhances performance across all datasets. Taking the generated frames as additional input to DUSt3R or MASt3R and selecting the most reliable prediction with our proposed self-consistency score outperforms relying on the input frame pair alone. This holds for all three video models for both rotation and translation.

On outward-facing datasets such as Cambridge Landmarks and ScanNet (Table 1), our method significantly

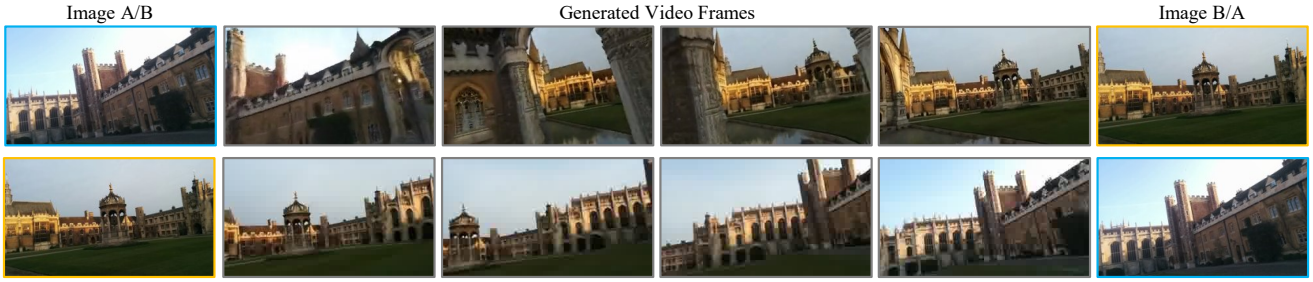


Figure 6. **Left-to-right bias.** We observed that video models exhibit a tendency to generate similar camera motions (e.g., both left-to-right pans) regardless of the intended direction of interpolation (i.e., transitioning from image A to image B or from image B to image A). This suggests an underlying bias within the model. To mitigate this bias, we swap the order of input images during the generation process.

reduces pose estimation errors. For example, when using our method with the DUSt3R pose estimator on Cambridge Landmarks, the mean rotation error decreases from 13.28° to 10.78° using Runway's model, while on ScanNet, mean rotation and translation errors decrease from (21.31°, 24.74°) to (17.65°, 15.88°) using Dream Machine. Many image pairs in these datasets feature outward-facing camera viewpoints with no overlap, which causes MASt3R to perform worse than DUSt3R, particularly on Cambridge Landmarks. In the supplementary material, we visualize scenarios where MASt3R fails completely. Despite this, our method still achieves improvements on both datasets when MASt3R is used as the pose estimator. Specifically, it reduces the mean rotation error from 36.55° to 27.47° on Cambridge Landmarks and increases the AUC at 30° from 55.10% to 58.28% on ScanNet when using video frames generated by Dream Machine.

On center-facing datasets such as DL3DV-10K and NAVI, the improvements are less pronounced but still present, as illustrated in Table 2, since these datasets inherently contain overlapping regions between input views. On the DL3DV-10K dataset, our method using the DUSt3R pose estimator reduces the mean translation error from 13.08° to 8.72°, and increases t_{acc}@30° from 89% to 94.67% using frames from Dream Machine. On the NAVI dataset, the DUSt3R pair only baseline already works well out of the box, but our method using video frames still reduces both mean rotation and translation errors by approximately 1° each. Because of the overlapping regions in these datasets, MASt3R benefits from the ability to leverage reliable matches, resulting in better performance than DUSt3R. Given the almost perfect performance of MASt3R on these datasets, our method, which takes video frames as additional input, achieves comparable results to MASt3R on the DL3DV-10K dataset when using only image pairs, and shows slight improvements on the NAVI dataset, decreasing the mean rotation and translation errors from (5.59°, 5.23°) to (5.28°, 5.20°) when using video frames generated by Runway.

We also evaluate an additional open-source video model CogVideoX-Interpolation [15, 63], ablate variants of our method, and evaluate the effectiveness of our method across different yaw changes in the supplement.

4.6. Qualitative Results

In Fig. 5, we visualize qualitative results of using DUSt3R on the input pairs alone compared with using selected generated frames from a video model. We find that all 3 video models are capable of generating informative intermediate images. We also visualize more video frames from all three video models in Fig. 4.

Please refer to the supplementary materials for more videos, interactive DUSt3R point clouds, and comparisons.

5. Conclusion

In this paper, we did a preliminary investigation into how a video model can be used to help pose estimation. We developed a heuristic for measuring the self-consistency of a generated video using a medoid-based selection algorithm, and we found that the additional context from the generated videos consistently helped a state-of-the-art pose estimator. This finding holds for the 3 recent publicly available video models that we were able to test. There is still significant room for improvement. That our oracle performs so much better than all other approaches reveals that finding a better video selection strategy is a fruitful area of research. We also found a number of limitations in current-generation video models. First, they are quite expensive and slow to run, which limited the scope of our investigation. Second, the videos still could not guarantee multi-view consistency. Although our medoid-distance-selection strategy helped alleviate this issue, sometimes all generated videos were low quality. Finally, we found that the video models are quite sensitive to minor changes such as prompts, camera intrinsics, and image aspect ratios.

Acknowledgments We would like to thank Keunhong Park, Matthew Levine, and Aleksander Hołynski for their feedback and suggestions.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023. 4

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *ICCV*, 2021. 7

[3] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A Space-Time Diffusion Model for Video Generation. In *SIGGRAPH Asia*, 2024. 2

[4] Axel Barroso-Laguna, Sowmya Munukutla, Victor Adrian Prisacariu, and Eric Brachmann. Matching 2D Images in 3D: Metric Relative Pose from Metric Correspondences. In *CVPR*, 2024. 2, 3

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *ECCV*, 2006. 2

[6] Hana Bezalel, Dotan Ankri, Ruojin Cai, and Hadar Averbuch-Elor. Extreme Rotation Estimation in the Wild. In *CVPR*, 2025. 2, 5

[7] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In *CVPR*, 2023. 2

[8] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 5

[9] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video Generation Models as World Simulators. 2024. 2

[10] Ruojin Cai, Bharath Hariharan, Noah Snavely, and Hadar Averbuch-Elor. Extreme Rotation Estimation using Dense Correlation Volumes. In *CVPR*, 2021. 2

[11] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-Baseline Relative Camera Pose Estimation with Directional Learning. In *CVPR*, 2021. 2

[12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. 5

[13] Emily Denton and Rob Fergus. Stochastic Video Generation with a Learned Prior. In *ICML*, 2018. 2

[14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *CVPRW*, 2018. 2

[15] Zhengcong Fei. CogVideoX-Interpolation: Keyframe Interpolation with CogVideoX, 2024. https://github.com/feizc/CogvideX-Interpolation [Accessed: (October 2024)]. 8

[16] Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 5

[17] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Fei-Fei Li, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic Video Generation with Diffusion Models. In *ECCV*, 2024. 2

[18] Richard I Hartley. In Defense of the Eight-Point Algorithm. *IEEE TPAMI*, 19(6):580–593, 1997. 1, 2

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *NeurIPS*, 2020. 2

[20] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen Video: High Definition Video Generation with Diffusion Models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[21] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video Diffusion Models. *NeurIPS*, 2022. 2

[22] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to Decompose and Disentangle Representations for Video Prediction. *NeurIPS*, 2018. 2

[23] Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan Popov, André Araujo, Ricardo Martin Brualla, Kaushal Patel, et al. NAVI: Category-Agnostic Image Collections with High-Quality 3D Shape and Pose Annotations. *NeurIPS*, 2023. 5

[24] Hanwen Jiang, Arjun Karpur, Bingyi Cao, Qixing Huang, and Andre Araujo. OmniGlue: Generalizable Feature Matching with Foundation Model Guidance. In *CVPR*, 2024. 2

[25] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image Matching across Wide Baselines: From Paper to Practice. *IJCV*, 2021. 7

[26] Arjun Karpur, Guilherme Perrotta, Ricardo Martin-Brualla, Howard Zhou, and André Araujo. LFM-3D: Learnable Feature Matching Across Wide Baselines Using 3D Signals. In *3DV*, 2024. 2

[27] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *ICCV*, 2015. 5

[28] Kuaishou. Kling AI, 2024. https://klingai.com/ [Accessed: (September 2024)]. 2

[29] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic Adversarial Video Prediction. *arXiv preprint arXiv:1804.01523*, 2018. 2

[30] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding Image Matching in 3D with MASt3R. In *ECCV*, 2024. 2, 3, 5

[31] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose++: Recovering 6D Poses from Sparse-view Observations. In *3DV*, 2024. 2

[32] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 2

[33] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A Large-Scale Scene Dataset for Deep Learning-based 3D Vision. In *CVPR*, 2024. 5

[34] H Christopher Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 293 (5828):133–135, 1981. 2

[35] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 2004. 1, 2, 5

[36] LumaAI. Luma Dream Machine, 2024. https://lumalabs.ai/dream-machine [Accessed: (September 2024)]. 2, 3, 7

[37] Marius Muja and David G Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *VISAPP*, 2(331-340):2, 2009. 2

[38] David Nistér. An Efficient Solution to the Five-point Relative Pose Problem. *IEEE TPAMI*, 26(6):756–770, 2004. 2

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2021. 2

[41] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *ICCV*, 2011. 2

[42] RunwayML. Tools for Human Imagination, 2024. https://runwayml.com/product [Accessed: (November 2024)]. 2, 3, 7

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *NeurIPS*, 2022. 2

[44] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal Generative Adversarial Nets with Singular Value Clipping. In *ICCV*, 2017. 2

[45] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*, 2020. 2

[46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *CVPR*, 2016. 3

[47] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *ECCV*, 2016. 3

[48] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-A-Video: Text-to-Video Generation without Text-Video Data. In *ICLR*, 2023. 2

[49] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning Using Nonequilibrium Thermodynamics. In *ICML*, 2015. 2

[50] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *NeurIPS*, 2018. 2

[51] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. *CVPR*, 2021. 2, 5

[52] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. QuadTree Attention for Vision Transformers. *ICLR*, 2022.

[53] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV*, 2020. 2

[54] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018. 2

[55] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning Local Features with Policy Gradient. *NeurIPS*, 2020. 2

[56] Ruben Villegas, Dumitru Erhan, Honglak Lee, et al. Hierarchical Long-term Video Prediction without Supervision. In *ICML*, 2018. 2

[57] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable Length Video Generation From Open Domain Textual Description. In *ICLR*, 2022. 2

[58] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating Videos with Scene Dynamics. *NeurIPS*, 2016. 2

[59] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: Solving Pose Estimation via Diffusion-aided Bundle Adjustment. In *ICCV*, 2023. 2, 7

[60] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *CVPR*, 2024. 1, 2, 3, 5

[61] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. CroCo v2: Improved Cross-view Completion Pre-training for Stereo Matching and Optical Flow. In *ICCV*, 2023. 3

[62] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. DynamiCrafter: Animating Open-domain Images with Video Diffusion Priors. In *ECCV*, 2024. 2, 3, 5

[63] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. *arXiv preprint arXiv:2408.06072*, 2024. 8

[64] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. RelPose: Predicting Probabilistic Relative Rotation for Single Objects in the Wild . In *ECCV*, 2022. 2

[65] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as Rays: Pose Estimation via Ray Diffusion. In *ICLR*, 2024. 2

[66] Qitao Zhao, Amy Lin, Jeff Tan, Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. DiffusionSfM: Predicting Structure and Motion via Ray Origin and Endpoint Diffusion. In *CVPR*, 2025. 2