AKiRa: Augmentation Kit on Rays for optical video generation

Supplementary Material

In this supplementary material, we include:

A. Additional quantitative results
B. Additional analysis of bokeh map
C. Ethical discussion
D. Algorithm of AKiRa
E. Details about user study
F. Discussion about metrics
G. Additional qualitative results

F. Additional quantitative results

In this section, we present additional quantitative results on the RealEstate10K dataset [71], comparing the performance of AKiRa with state-of-the-art camera control approaches for video generation: MotionCtrl [56] and CameraCtrl [21], incorporating the corresponding LoRA [26] module to ensure domain consistency.

We evaluate video quality, camera motion fidelity, and temporal dynamic consistency metrics (same as in Table 1 of the main manuscript) for the text-to-video (T2V) backbone Animatediff [18] and the image-to-video (I2V) backbone SVD [5].

All metrics are computed on 1,000 generated samples using text prompts from the RealEstate10K dataset or conditioning frames from the WebVid dataset [3]. For AKiRa, as it explicitly controls the bokeh, we set $\alpha = 0$ to align with the aperture behavior of the RealEstate10K dataset, where videos are recorded using small apertures and wide-angle cameras. The impact of aperture on performance is discussed in Section G and detailed in Table 6.

In Table 6, we present the performance of AKiRa in comparison with two state-of-the-art methods: MotionCtrl [56] and CameraCtrl [21]. In terms of video quality, AKiRa outperforms both SoTA methods on FVD and CD-FVD metrics for both the Animatediff and SVD backbones, achieving scores of 128.55 and 89.16 for Animatediff, and 54.83 and 41.55 for SVD. A similar trend is observed in dynamic consistency, where AKiRa leads across all metrics on both backbones. It achieves the highest scores for Consistency (0.9851), Smoothness (0.9933), and Flickering (0.9733), demonstrating superior temporal coherence. In terms of motion fidelity, AKiRa demonstrates significantly better motion controllability on the Animatediff backbone, achieving superior performance as evidenced by the highest FlowSim and lowest RPE errors. AKiRa competes closely with CameraCtrl on SVD, with only a narrow difference. We attribute this to the overfitting of CameraCtrl on the real-estate dataset, where intrinsic parameters and optical features remain unchanged during this experiment.

G. Additional analysis of bokeh map

Controllability of Bokeh - Aperture We propose a bokeh map with the same structure and dimensions as the direction and moment maps, assigning an aperture and focus (depth-of-field) parameter to each pixel in the frame. More specifically, we define the coordinates of the focus point $(u_{\text{in}}, v_{\text{in}})$ representing the sharpest point in the frame. We then define the per-pixel bokeh map as $\mathbf{a} \in \mathbb{R}^3$ for any point (u, v) on the frame as:

$$\mathbf{a} = \begin{bmatrix} u - u_{\text{in}} \\ v - v_{\text{in}} \\ \|(u, v) - (u_{\text{in}}, v_{\text{in}})\| \frac{1}{\sigma(\alpha)} \end{bmatrix}, \tag{7}$$

We first present the visualization of the bokeh map and demonstrate how it influences the generated videos. We examine two groups of bokeh variations: the effect of varying the aperture α in Figure 7a, and the effect of adjusting the focus point f_{in} in Figure 7b (see the zoomed image in the second row highlighted the red rectangle).

In Figure 7a, we progressively increase the aperture level over time, with the focus fixed at the center of the image. As a result, the visualization of the bokeh map shrinks, and it shows that the blur area expands proportionally with increasing aperture α.

In Figure 7b, we shift the focus point f_{in} from the upper-left corner to the lower-right corner. This causes the center of the bokeh map to move accordingly, effectively shifting the blur area in generated videos. The results confirm our ability to dynamically control the blur area based on the focus point.

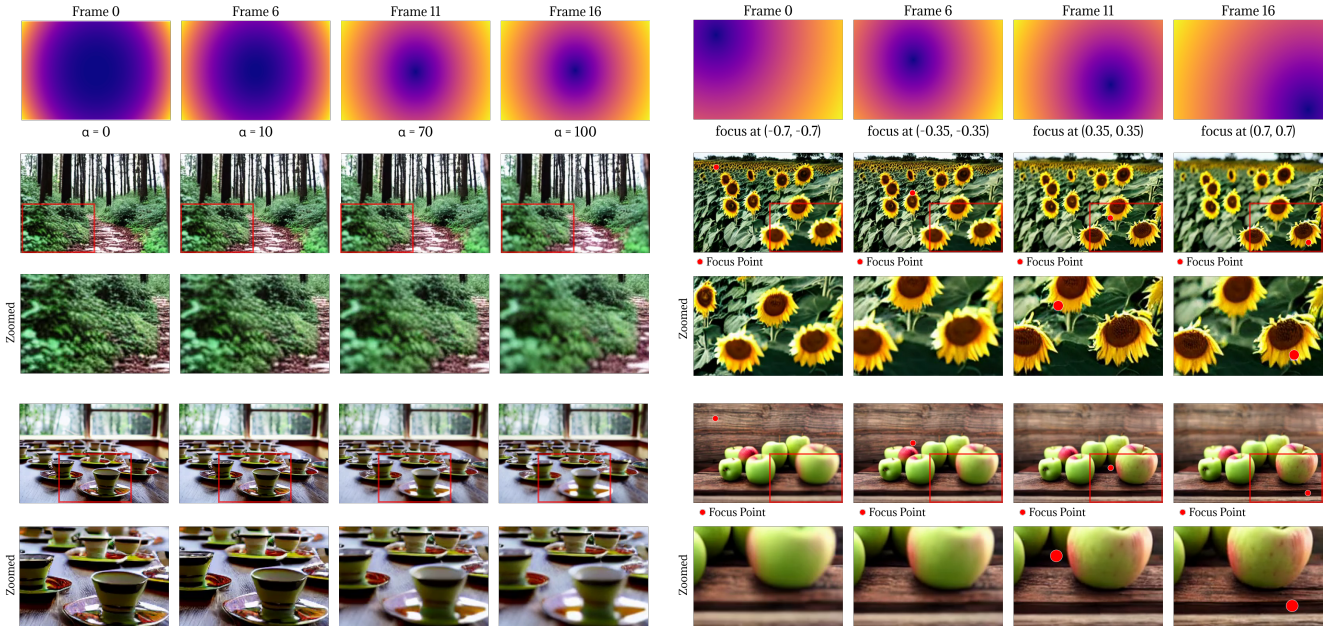
Both experiments validate the effectiveness and controllability of AKiRa in manipulating the depth of field.

Bokeh influence on T2V performance We analyze the influence of bokeh on video quality, flow similarity, and dynamic consistency using Animatediff, with the experimental settings identical to those in Table 1 of the main manuscript. By varying the aperture value α from 0 to 100, we measure its impact on the corresponding metrics.

Table 6 reports the results for different aperture settings. When the aperture is small and bokeh is weak (i.e., $\alpha = 0$), the generated videos exhibit better consistency. Conversely, when the aperture is large (i.e., $\alpha = 100$), the videos demonstrate greater smoothness and reduced flickering. Notably, the optimal video generation metrics are observed around $\alpha = 30$ and $\alpha = 50$. This can be attributed to **the intrinsic bokeh** present in the WebVid [3] dataset, where **adding an appropriate amount of bokeh enhances**

Method		Video quality		Camera motion fidelity			Dynamic consistency (VBench)		
Backbone	Camera control	FVD ↓	CD-FVD ↓	RPE-R (deg)↓	RPE-t (cm)↓	FlowSim ↑	Consistency ↑	Smoothness ↑	Flickering↑
AnimateDiff [18]	MotionCtrl [56]	237.22	543.24	0.387	1.536	67.83	0.9779	0.9834	0.9712
	CameraCtrl [21]	177.70	106.24	0.377	1.555	77.08	0.9779	0.9834	0.9712
	AKiRa (ours)	128.55	89.16	0.323	1.347	84.04	0.9809	0.9882	0.9745
SVD [5]	MotionCtrl [56]	122.67	330.75	1.030	1.326	24.14	0.9516	0.9814	0.9404
	CameraCtrl [21]	55.55	50.18	0.312	1.268	92.19	0.9836	0.9928	0.9695
	AKiRa (ours)	54.83	41.55	0.312	1.236	91.51	0.9851	0.9933	0.9733

Table 5. **Comparison with the state-of-the-art.** Comparison of AKiRa and concurrent methods with different backbones on RealEstate dataset, evaluating video quality, camera motion fidelity, and dynamic consistency. Best.



(a) Example of increasing apertures along video time

(b) Example of shifting focus point (red dot) along video time

Figure 7. **Qualitative Results.** We demonstrate the qualitative performance of AKiRa on bokeh variations for both (a) aperture levels and (b) focus points. The results are generated using Animatediff [18]. In (a), we observe that the blur area and intensity increase proportionally with the aperture parameter α. In (b), the blur area shifts dynamically following changes in the focus point, showing the effectiveness of AKiRa in handling variations in depth of field.

Apert.	Video quality		Motion fidelity	Dynamic consistency (VBench)		
	FVD ↓	CD-FVD ↓	FlowSim ↑	Consistency ↑	Smoothness ↑	Flickering↑
0	350.05	333.61	70.82	0.9697	0.9702	0.9525
5	353.09	337.48	70.94	0.9695	0.9705	0.9528
10	354.94	334.95	71.15	0.9692	0.9709	0.9534
30	341.31	327.02	71.23	0.9686	0.9728	0.9559
50	332.27	328.74	70.97	0.9686	0.9735	0.9572
100	342.77	328.13	70.82	0.9683	0.9737	0.9574

Table 6. **Influence of aperture.** Influence of aperture effect on AKiRa's performances with Animatediff backbone on RealEstate dataset. Best.

realism, resulting in improved FVD and CD-FVD performance.

H. Ethical discussion

Our paper proposes a method to generate videos based on camera and optical control, enabling better alignment of the generation process with user intentions and geometric information. On the positive side, this approach can enhance the AIGC (AI-Generated Content) creative process by reducing biases introduced by training data, more accurately reflecting user intentions, and minimizing trial-and-error in content generation. This efficiency can also contribute to reducing the carbon footprint associated with the generation process. On the negative side, however, it may reduce the labour required for video production, potentially leading to job losses, and could stifle creativity if individuals become overly reliant on generative tools.

I. Algorithm of AKiRa

We present the complete AKiRa algorithm in Algorithm 1. As discussed in the main manuscript, random sampling is implemented using spline sampling. Augmentation dropout with a probability p is applied to all the optical features, both collectively and individually.

Since augmentation is performed on-the-fly during the training process, the augmentation order is carefully designed to optimize computational efficiency. Specifically, we first augment the bokeh aperture to leverage the precomputed depth map derived from the original frames. Next, distortion augmentation is applied, which may implicitly alter the focal length due to a necessary cropping operation to avoid undefined borders during image warping. Finally, we augment the zoom aspect, incorporating the results of the distortion augmentation.

During training, the augmentation parameters are sampled as follows: the dropout probability p is set to 0.2; the bokeh aperture is sampled uniformly between 0 and 100; the distortion parameters are sampled uniformly within the range $[-0.1, 0.1]$ for all three parameters in $\mathbf{D}$; and the zoom factor is sampled between 1.0 and 3.0, as zoom factors below 1.0 are ill-defined due to the difficulty to generate outpainting content through augmentation.

Algorithm 1 AKiRa augmentation algorithm

Require: $\mathbf{I}$: frames, $\mathbf{Z}$: depth maps, p: aug. dropout.
  if True with probability p **then**
    if True with probability p **then**
      $\{\alpha\} = \text{RANDOMAPERTURESPLINE}()$
      $\{(u, v)\} = \text{RANDOMINFOCUSSPLINE}()$
      $\mathbf{I} \leftarrow \text{BOKEHAUGMENTER}(\mathbf{I}, \mathbf{Z}, \{\alpha\}, \{(u, v)\})$
    end if
    if True with probability p **then**
      $\{\mathbf{D}\} = \text{RANDOMDISTORTIONSPLINE}()$
      $\mathbf{I} \leftarrow \text{DISTORTIONAUGMENTER}(\mathbf{I}, \{\mathbf{D}\})$
    end if
    if True with probability p **then**
      $\{f\} = \text{RANDOMFOCALSPLINE}()$
      $\mathbf{I} \leftarrow \text{ZOOMAUGMENTER}(\mathbf{I}, \{\text{f}\})$
    end if
    return $\mathbf{I}$
  else return $\mathbf{I}$
  end if

J. Details about user study

In this section, we elaborate on the specifics of our user study setup corresponding to Section 4.2 in our main manuscript.

For the evaluation, each participant was presented with a total of 10 video sets, 5 with Animatediff backbone (T2V) and 5 with SVD backbone (I2V). Each set comprised 4 generated videos from (i) the baseline (backbone without camera control), (ii) MotionCtrl, (iii) CameraCtrl, and (iv) AKiRa (ours), we shuffled the results and displayed them in random order.

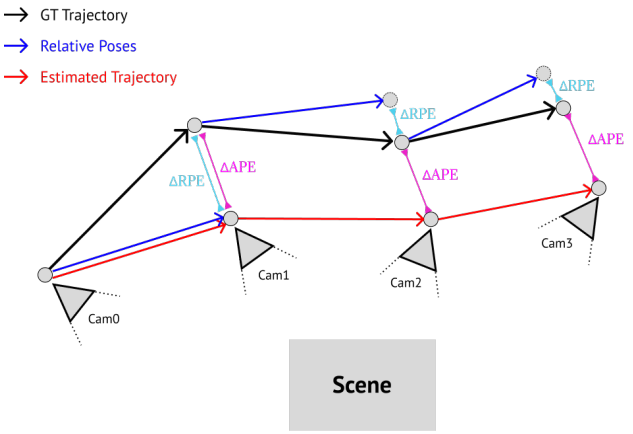


Figure 8. **Illustrates the instability of Absolute Pose Error (APE), purple** when measuring trajectory accuracy, compared to the robustness of Relative Pose Error (RPE), cyan. APE tends to accumulate errors, especially at later frames. In contrast, RPE calculates errors based on relative transformations between consecutive frames, making it less sensitive to single-frame errors and more robust for trajectory evaluation.

Subsequently, participants were prompted with 6 questions for each comparison:

1. *Rank the consistency of the video with the text prompt* (Only for Animatediff backbone).
2. *Rank the video quality (i.e. temporal consistency).*
3. *Rank the camera motion consistency with the reference.*
4. *Rank the best zoom-in or -out effect.*
5. *Rank the best distortion effect (e.g. fisheye).*
6. *Rank the best bokeh (in- or out-of-focus effect).*

In total, we recorded 25 participants with each participant responding to 55 questions. We analyzed the results by examining responses to each question individually, summarizing the collective feedback.

K. Discussion about metrics

K.1. Drawbacks of SfM-based metrics

Absolute Pose Error Many recent works [21, 56, 58] on camera motion-controlled video generation rely on Structure-from-Motion (SfM) or SLAM-based metrics to evaluate the effectiveness of camera control. The primary metric utilized in these works is similar to the Absolute Pose Error (APE) used in SLAM and SfM applications [43, 48], which computes the average trajectory error for translation and rotation separately. These errors are defined as follows:

For translation error in $\mathbb{R}^3$:

$$\text{APE}_{\text{trans}} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathbf{t}}_i - \mathbf{t}_i^*\|,$$

where N is the total number of frames, $\hat{\mathbf{t}}_i \in \mathbb{R}^3$ is the estimated camera translation vector for frame i, and $\mathbf{t}_i^* \in \mathbb{R}^3$ is the ground truth translation vector for the same frame.

For rotation error in $SO(3)$, the error is often computed as the angle of the relative rotation:

$$\text{APE}_{\text{rot}} = \frac{1}{N}\sum_{i=1}^{N} \arccos\left(\frac{\text{trace}\left(\hat{\mathbf{R}}_i(\mathbf{R}_i^*)^\top\right) - 1}{2}\right),$$

where $\hat{\mathbf{R}}_i \in SO(3)$ is the estimated rotation matrix for frame i, and $\mathbf{R}_i^* \in SO(3)$ is the ground truth rotation matrix for the same frame. However, APE is highly sensitive to errors in individual frames, which is a common issue in generated videos due to **flickers or sudden object movements**. These artifacts, often **irrelevant to the quality of motion control**, can disproportionately affect the evaluation.

Relative Pose Error To address the APE limitation, related domains often rely on the Relative Pose Error (RPE), which reduces the impact of **accumulated errors** caused by single-frame inaccuracies, especially when such errors occur at the early stages of the trajectory.

RPE is computed by comparing the relative transformations between consecutive frames, rather than the absolute poses. It is defined separately for translation and rotation as follows:

For translation error in $\mathbb{R}^3$:

$$\text{RPE}_{\text{trans}} = \frac{1}{N-1}\sum_{i=1}^{N-1} \|\Delta\hat{\mathbf{t}}_i - \Delta\mathbf{t}_i^*\|,$$

where $\Delta\hat{\mathbf{t}}_i = \hat{\mathbf{t}}_{i+1} - \hat{\mathbf{t}}_i$ is the estimated relative translation, and $\Delta\mathbf{t}_i^* = \mathbf{t}_{i+1}^* - \mathbf{t}_i^*$ is the ground truth relative translation.

For rotation error in $SO(3)$, the relative error is defined as:

$$\text{RPE}_{\text{rot}} = \frac{1}{N-1}\sum_{i=1}^{N-1} \arccos\left(\frac{\text{trace}\left(\Delta\hat{\mathbf{R}}_i\Delta\mathbf{R}_i^*\right) - 1}{2}\right),$$

where $\Delta\hat{\mathbf{R}}_i = \hat{\mathbf{R}}_{i+1}\hat{\mathbf{R}}_i^\top$ is the estimated relative rotation, and $\Delta\mathbf{R}_i^* = \mathbf{R}_{i+1}^*\mathbf{R}_i^{*\top}$ is the ground truth relative rotation.

We demonstrate this phenomenon in Figure 8. When the first frame exhibits a high APE (purple), even if the subsequent trajectory is relatively accurate, the error is accumulated throughout the trajectory. In contrast, RPE is computed between relative poses (cyan), **making it less biased by errors in previous estimations**, therefore providing a more robust assessment of motion control quality.

Scaling Ambiguity Another challenge when using 3D metric errors to evaluate video camera control quality arises from unknown intrinsic parameters. Similar camera motions in image frames can have different interpretations in 3D metrics; for example, a leftward motion with similar visual displacement can correspond to varying metric distances depending on the scale of the scene [19]. While some Structure-from-Motion (SfM) methods estimate camera intrinsics, these estimates are often unreliable due to limited frame numbers (often around 15) and the typically smooth nature of camera motion, with a short stereo baseline needed for accurate intrinsic estimation.

To address this issue, in our paper, we report the **scale-corrected** camera trajectory by normalizing the trajectory length to match the ground truth. Formally, this is done by rescaling the estimated trajectory $\hat{\mathbf{T}}$ such that:

$$\hat{\mathbf{T}}_{\text{scaled}} = \frac{\|\mathbf{T}^*\|}{\|\hat{\mathbf{T}}\|} \cdot \hat{\mathbf{T}},$$

where $\|\mathbf{T}^*\|$ is the length of the ground truth trajectory and $\|\hat{\mathbf{T}}\|$ is the length of the estimated trajectory.

In our paper, all computations are performed using `evo`[1], a standard trajectory evaluation toolbox widely used for SLAM and visual odometry evaluations.

Computational Efficiency Unfortunately, the computation time for SfM is relatively long and, most importantly, difficult to parallelize on GPU due to the sequential nature of the optimization problem. For instance, processing a single video with 16 frames using ParticleSfM [67] can take up to 4 minutes on average, including feature extraction and the optimization pipeline required for convergence. In our study, computing SfM for 1000 generated videos required an average of 66 hours on a single CPU.

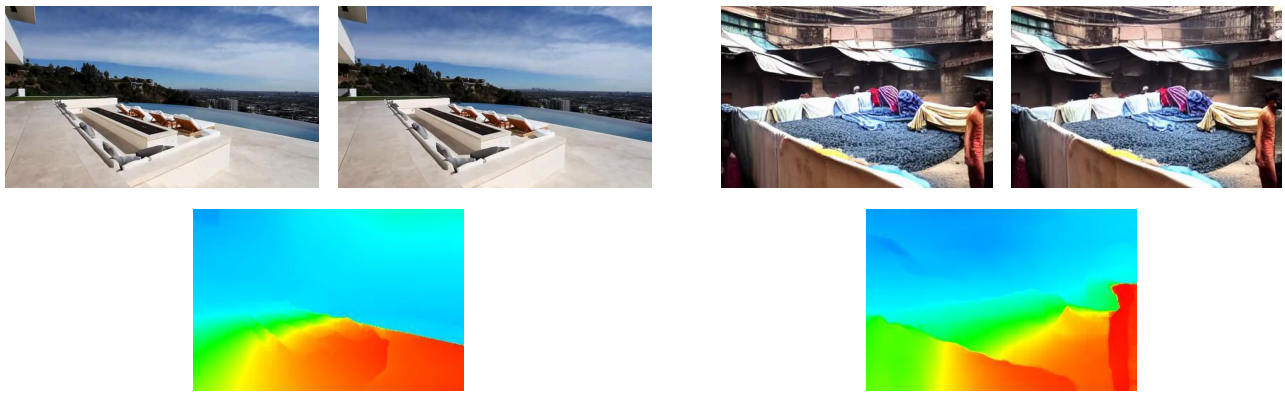
K.2. FlowSim metric

Flow Similarity. As detailed in Section 4.1 and Equation 6 of the main manuscript, we introduce the flow similarity metric. Similar to RPE, the concept of optical flow involves computing the relative on-image pixel motion between frames, which is widely used as an intermediate feature in SLAM and dense SfM processes [9, 66].

Compared to other on-image features, optical flow is less content-dependent and can serve as a robust metric for comparing video similarity. Unlike SfM trajectory features, optical flow does not rely on prior knowledge or precise estimation of the ground truth scaling of the scene. This makes it inherently robust to scaling ambiguities.

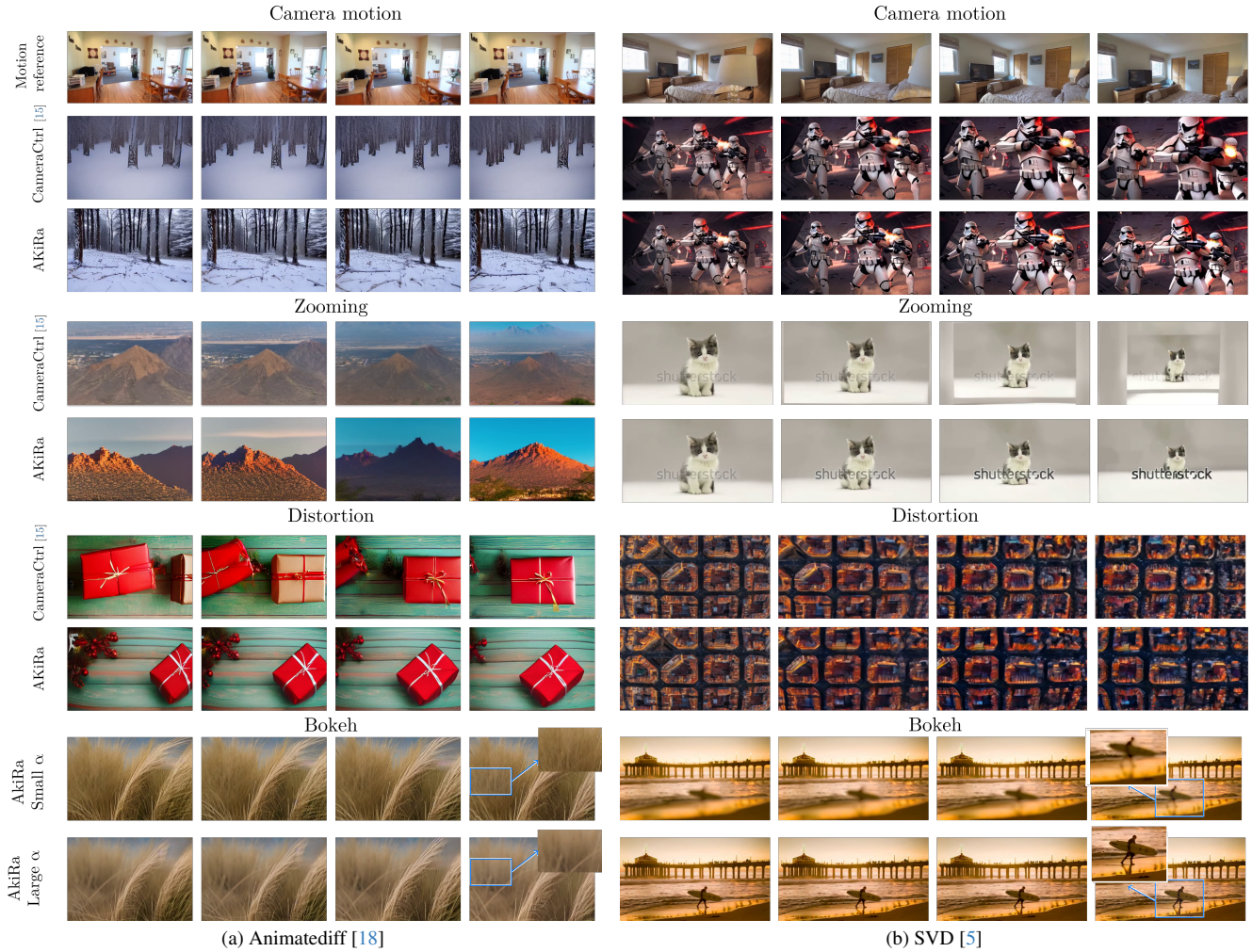
In our implementation, we measure the alignment of two optical flow directions while ignoring magnitudes to avoid content bias, as magnitudes are often influenced by

[1] https://github.com/MichaelGrupp/evo



(a) Ground-truth optical flow. (b) Generated optical flow

Figure 9. **Comparison of optical flow** for different content and similar camera motion.



(a) Animatediff [18] (b) SVD [5]

Figure 10. **Qualitative results** of AKiRa on Animatediff [18] and SVD [5] backbones. We recommend viewing the supplementary video.

disparity (i.e., differences in depth). Additionally, we exclude low-magnitude components because their directions are typically unreliable.

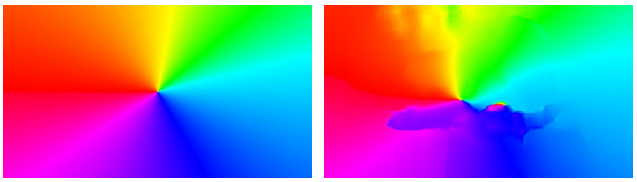
Figure 9 highlights the robustness of optical flow in com-

paring camera motion between videos. Despite differing content, similar flows indicate similar camera motion.

Zoom flow similarity. To evaluate the quality of the zooming effect, we compute the flow similarity between the theoretical zooming flow and the generated one. The theoretical zooming flow is derived from Equation 5 in the main manuscript as:

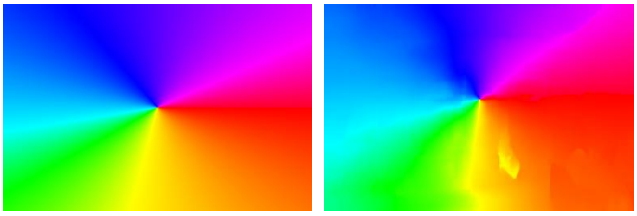
$$\begin{bmatrix} u_{\text{fl}} \\ v_{\text{fl}} \end{bmatrix} - \begin{bmatrix} u_{\text{fl}'} \\ v_{\text{fl}'} \end{bmatrix} = (s - s') \begin{bmatrix} u - c_x \\ v - c_y \end{bmatrix} , \tag{8}$$

where s and s' denote the zooming scales, and (c_x, c_y) are the principal point coordinates (i.e. the screen center). As illustrated in Figure 11a, the flow direction for a zoom-in effect converges toward the principal point (or diverges outward for a zoom-out effect), aligning with Equation 8. The generated flow, shown in Figure 11b, closely matches the theoretical flow, with minor deviations caused by variations in frame content.



(a) Theoretical flow (b) Generated flow

Figure 11. **Theoretical vs. generated zooming flows.**



(a) Theoretical flow (b) Generated flow

Figure 12. **Theoretical vs. generated distorted flows.**

Distortion flow similarity. To evaluate the quality of the distortion effect, we compute the flow similarity between the theoretical distortion flow and the generated one. The theoretical distortion flow is derived from Equation 2 in the main manuscript as:

$$\begin{bmatrix} u_{\mathbf{D}} \\ v_{\mathbf{D}} \end{bmatrix} - \begin{bmatrix} u_{\mathbf{D}'} \\ v_{\mathbf{D}'} \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} (\mathbf{D} - \mathbf{D}') \begin{bmatrix} r^2 \\ r^4 \\ r^6 \end{bmatrix} , \tag{9}$$

where $\mathbf{D}$ and $\mathbf{D}'$ denote the distortion parameters and $r = \sqrt{(u - c_x)^2 + (v - c_y)^2}$ distance of each pixel towards image center. As illustrated in Figure 12a, the flow direction for a distortion effect also converges toward the principal point —or diverges outward depending on the sign of $\mathbf{D} - \mathbf{D}'$—, aligning with Equation 9. The generated flow, shown in Figure 12b, closely matches the theoretical flow, again, with minor deviations caused by variations in frame content.

Computational Efficiency During implementation, we use RAFT [49], a fast deep optical flow estimator with GPU-based implementation, which significantly speeds up the flow estimation process compared to SfM and can be easily parallelized on GPU. For example, computing optical flows for 1000 generated videos takes approximately 10 minutes on a GPU, compared to 66 hours required by SfM on a single CPU.

As a result, our key messages concerning the evaluation metrics are:

1. Pose metrics for assessing camera control in generated videos are intrinsically less accurate, particularly when directly computing APE.
2. Using RPE with scale correction techniques improves robustness; however, the computational cost is prohibitively high for scaling to AI-generated videos.
3. We propose optical flow similarity (FlowSim), based on flow direction, offers a viable alternative. It serves as a good approximation of RPE while being computationally efficient, fast to compute, and scalable for large-scale AI-generated videos.
4. The flow similarity metric can also be used to confirm other optic features than motion, such as focal length change (zoom) and distortions.

L. Additional qualitative results

We provide additional qualitative results in Figure 10, demonstrating that AKiRa method performs well in several key aspects: accurately capturing camera motion (trees) with high video quality (the middle stormtrooper's head cf. CameraCtrl); maintaining consistency during zooming (mountain peak, cat's frame); effectively reflecting distortion effects (gift, bird's-eye view of Barcelona city); and rendering various aperture and bokeh effects (surfing on the beach, grass) with controllability.

References

[1] An, J., Zhang, S., Yang, H., Gupta, S., Huang, J.B., Luo, J., Yin, X.: Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. arXiv preprint arXiv:2304.08477 (2023) 2

[2] Bahmani, S., Skorokhodov, I., Siarohin, A., Menapace, W., Qian, G., Vasilkovsky, M., Lee, H.Y., Wang, C., Zou, J., Tagliasacchi, A., et al.: Vd3d: Taming

large video diffusion transformers for 3d camera control. arXiv preprint arXiv:2407.12781 (2024) 3

[3] Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: ICCV (2021) 2, 7, 10

[4] Bertalmío, M.: Image processing for cinema. CRC Press (2014) 3

[5] Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023) 2, 3, 5, 7, 8, 10, 11, 14

[6] Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: CVPR (2023) 2

[7] Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., Yang, S., Xing, J., Liu, Y., Chen, Q., Wang, X., et al.: Videocrafter1: Open diffusion models for high-quality video generation. arXiv preprint arXiv:2310.19512 (2023) 2

[8] Chen, W., Liu, F., Wu, D., Sun, H., Song, H., Duan, Y.: Dreamcinema: Cinematic transfer with free camera and 3d character. arXiv preprint arXiv:2408.12601 (2024) 3

[9] Cheng, J., Sun, Y., Meng, M.Q.H.: Improving monocular visual slam in dynamic environments: an optical-flow-based approach. Advanced Robotics (2019) 13

[10] Cheong, S.Y., Ceylan, D., Mustafa, A., Gilbert, A., Huang, C.H.P.: Boosting camera motion control for video diffusion transformers. arXiv preprint arXiv:2410.10802 (2024) 3

[11] Courant, R., Dufour, N., Wang, X., Christie, M., Kalogeiton, V.: E.T. the Exceptional Trajectories: text-to-camera-trajectory generation with character awareness. In: ECCV (2024) 3

[12] Courant, R., Lino, C., Christie, M., Kalogeiton, V.: High-level features for movie style understanding. In: ICCV-W (2021) 7

[13] Davtyan, A., Sameni, S., Favaro, P.: Efficient video prediction via sparsely conditioned flow matching. In: ICCV (2023) 2

[14] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS (2021) 2

[15] Dufour, N., Picard, D., Kalogeiton, V.: Scam! transferring humans between images with semantic cross attention modulation. In: ECCV. Springer (2022) 3

[16] Gauthier, J.: Conditional generative adversarial nets for convolutional face generation. Class project for Stanford CS231N: convolutional neural networks for visual recognition, Winter semester (2014) 3

[17] Ge, S., Mahapatra, A., Parmar, G., Zhu, J.Y., Huang, J.B.: On the content bias in fréchet video distance. In: CVPR (2024) 6

[18] Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In: ICLR (2023) 2, 3, 5, 7, 8, 10, 11, 14

[19] Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004) 3, 13

[20] Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003) 3

[21] 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) 2, 3, 4, 6, 7, 8, 10, 11, 12

[22] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017) 6

[23] Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022) 2

[24] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020) 2

[25] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. NeurIPS (2022) 2

[26] Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: ICLR (2022) 3, 10

[27] Hu, T., Zhang, J., Yi, R., Wang, Y., Huang, H., Weng, J., Wang, Y., Ma, L.: Motionmaster: Training-free camera motion transfer for video generation. arXiv preprint arXiv:2404.15789 (2024) 3

[28] Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: CVPR (2024) 6

[29] Jiang, H., Christie, M., Wang, X., Liu, L., Wang, B., Chen, B.: Camera keyframing with style and control. ACM TOG (2021) 3

[30] Jiang, H., Wang, B., Wang, X., Christie, M., Chen, B.: Example-driven virtual cinematography by learning camera behaviors. ACM TOG (2020) 3

[31] Jiang, H., Wang, X., Christie, M., Liu, L., Chen, B.: Cinematographic camera diffusion model. In: Computer Graphics Forum (2024) 3

[32] Jiang, X., Rao, A., Wang, J., Lin, D., Dai, B.: Cinematic behavior transfer via nerf-based differentiable filming. In: CVPR (2024) 3

[33] Jinyu, L., Bangbang, Y., Danpeng, C., Nan, W., Guofeng, Z., Hujun, B.: Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality. Virtual Reality & Intelligent Hardware (2019) 6

[34] Ma, X., Wang, Y., Jia, G., Chen, X., Liu, Z., Li, Y.F., Chen, C., Qiao, Y.: Latte: Latent diffusion transformer for video generation. arXiv preprint arXiv:2401.03048 (2024) 2

[35] Menapace, W., Siarohin, A., Skorokhodov, I., Deyneka, E., Chen, T.S., Kag, A., Fang, Y., Stoliar, A., Ricci, E., Ren, J., et al.: Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. In: CVPR (2024) 2

[36] Mildenhall, B., Srinivasan, P., Tancik, M., Barron, J., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 3

[37] Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. NeurIPS (2016) 3

[38] Pan, L., Baráth, D., Pollefeys, M., Schönberger, J.L.: Global structure-from-motion revisited. In: ECCV (2024) 6

[39] Peng, J., Cao, Z., Luo, X., Lu, H., Xian, K., Zhang, J.: Bokehme: When neural rendering meets classical rendering. In: CVPR (2022) 6

[40] Plücker, J.: Analytisch-geometrische Entwicklungen. GD Baedeker (1828) 2, 4

[41] Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.Y., Chuang, C.Y., et al.: Movie gen: A cast of media foundation models. arXiv preprint arXiv:2410.13720 (2024) 2

[42] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 2

[43] Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 6, 12

[44] Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022) 2

[45] Slama, C.C., Theurer, C., Henriksen, S.W.: Manual of photogrammetry. American Society of Photogrammetry (1980) 4

[46] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015) 2

[47] Song, Y., Ermon, S.: Improved techniques for training score-based generative models. NeurIPS (2020) 2

[48] Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems (2012) 6, 12

[49] Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV (2020) 6, 15

[50] Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018) 6

[51] Wang, J., Yuan, H., Chen, D., Zhang, Y., Wang, X., Zhang, S.: Modelscope text-to-video technical report. arXiv preprint arXiv:2308.06571 (2023) 2

[52] Wang, X., Courant, R., Shi, J., Marchand, E., Christie, M.: Jaws: just a wild shot for cinematic transfer in neural radiance fields. In: CVPR (2023) 3

[53] Wang, X., Yuan, H., Zhang, S., Chen, D., Wang, J., Zhang, Y., Shen, Y., Zhao, D., Zhou, J.: Videocomposer: Compositional video synthesis with motion controllability. NeurIPS (2024) 3

[54] Wang, Y., Chen, X., Ma, X., Zhou, S., Huang, Z., Wang, Y., Yang, C., He, Y., Yu, J., Yang, P., et al.: Lavie: High-quality video generation with cascaded latent diffusion models. arXiv preprint arXiv:2309.15103 (2023) 2

[55] Wang, Z., Li, Y., Zeng, Y., Fang, Y., Guo, Y., Liu, W., Tan, J., Chen, K., Xue, T., Dai, B., et al.: Humanvid: Demystifying training data for camera-controllable human image animation. arXiv preprint arXiv:2407.17438 (2024) 3

[56] Wang, Z., Yuan, Z., Wang, X., Li, Y., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. In: SIGGRAPH (2024) 2, 3, 6, 7, 8, 10, 11, 12

[57] Wu, J., Li, X., Zeng, Y., Zhang, J., Zhou, Q., Li, Y., Tong, Y., Chen, K.: Motionbooth: Motion-aware customized text-to-video generation. CoRR (2024) 3

[58] Xu, D., Nie, W., Liu, C., Liu, S., Kautz, J., Wang, Z., Vahdat, A.: Camco: Camera-controllable 3d-consistent image-to-video generation. arXiv preprint arXiv:2406.02509 (2024) 3, 12

[59] Ya, G., Favero, G., Luo, Z.H., Jolicoeur-Martineau, A., Pal, C., et al.: Beyond fvd: Enhanced evaluation metrics for video generation quality. arXiv preprint arXiv:2410.05203 (2024) 6

[60] Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. arXiv preprint arXiv:2406.09414 (2024) 6

[61] Yang, S., Hou, L., Huang, H., Ma, C., Wan, P., Zhang, D., Chen, X., Liao, J.: Direct-a-video: Customized

video generation with user-directed camera movement and object motion. In: SIGGRAPH (2024) 3

[62] Yang, Y., Lin, H., Yu, Z., Paris, S., Yu, J.: Virtual dslr: High quality dynamic depth-of-field synthesis on mobile platforms. Electronic Imaging (2016) 6

[63] Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721 (2023) 3

[64] Zhang, J.Y., Lin, A., Kumar, M., Yang, T.H., Ramanan, D., Tulsiani, S.: Cameras as rays: Pose estimation via ray diffusion. In: ICLR (2024) 2, 4

[65] Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023) 3

[66] Zhang, T., Zhang, H., Li, Y., Nakamura, Y., Zhang, L.: Flowfusion: Dynamic dense rgb-d slam based on optical flow. In: ICRA. IEEE (2020) 13

[67] Zhao, W., Liu, S., Guo, H., Wang, W., Liu, Y.J.: Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In: ECCV (2022) 6, 13

[68] Zhao, W., Wei, F., Wang, H., He, Y., Lu, H.: Full-scene defocus blur detection with defbd+ via multi-level distillation learning. IEEE Transactions on Multimedia (2023) 7

[69] Zheng, G., Li, T., Jiang, R., Lu, Y., Wu, T., Li, X.: Cami2v: Camera-controlled image-to-video diffusion model. arXiv preprint arXiv:2410.15957 (2024) 3

[70] Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all (2024), https://github.com/hpcaitech/Open-Sora 2

[71] Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. ACM TOG (2018) 3, 8, 10