RealCam-I2V: Real-World Image-to-Video Generation with Interactive Complex Camera Control

Appendix

A. Dataset

The camera trajectory of each video clip from RealEstate10K [101] is first derived by SLAM methods at lower resolution with the field of view fixed at 90°. The authors then refine each of camera sequence using a structure-from-motion (SfM) pipeline, performing feature extraction, feature matching and global bundle adjustment successively. Given the unawareness of global scene scale, the resulted camera poses of RealEstate10K are up to an arbitrary scale per clip. For each frame the authors compute the 5-th percentile depth among all point depths from that frame's camera. Computing this depth across all cameras in a video clip gives a set of near plane depths and the whole scene is scaled so that the 10-th percentile of this set of depths is 1.25m.

While using RealEstate10K's scenes and camera trajectories during inference avoids scale issues within the dataset, challenges arise in more general cases. Specifically, when pairing out-of-domain images with either in-domain or out-of-domain trajectories, the inconsistencies between training and inference scales become evident. *These inconsistencies make it impossible to generate realistic and controllable videos.*

The solution lies in reconstructing an absolute-scale scene for any given image. By leveraging metric depth predictor, we can reconstruct the absolute-scale 3D scene for the reference image. This absolute-scale scene bridges the gap between training and inference, enabling robust generalization to real-world applications. With this alignment, the model becomes capable of handling diverse combinations of images and trajectories, ensuring consistent and reliable performance across various scenarios.

B. Training

We choose DynamiCrafter [76] as our image-to-video (I2V) base model. We trained proposed method on 4 publicly accessible variants of DynamiCrafter, namely `256`, `512`, `512_interp` and `1024`. We conduct ablation study on resolution 256×256, due to the limitation of computing resource. For resolution 256×256, we train all models on ϵ-prediction with effective batch size 64 for 50,000 steps, taking about 50 hours. For resolution 512×320 and 1024×576, we train RealCam-I2V on v-prediction while enable `perframe_ae` and `gradient_checkpoint` to reduce peak GPU memory consumption. We apply the Adam optimizer with a constant learning rate of 1×10^{-4} with mixed-precision fp16 and ZeRO-1.

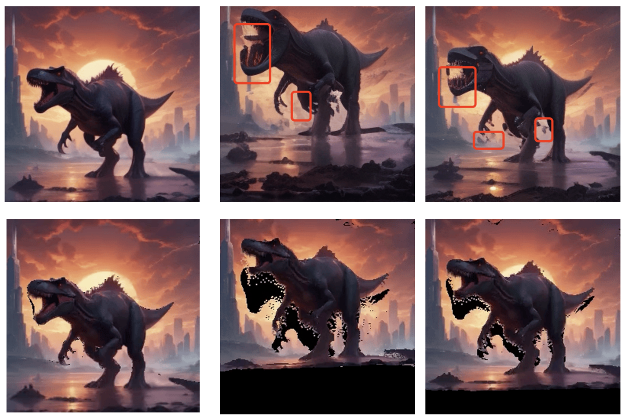
Figure 11. **Outlier filtering for noise shaping.** Undesired pixels will not be pasted in mistake for filtering kernel size $k \geq 3$.

For MotionCtrl [69] and CameraCtrl [22], we reproduce all results on DynamiCrafter for fair comparison. For CamI2V [97], we implement hard mask epipolar attention and set 2 register tokens, aligned with the original paper. In quantitative comparison and ablation study, we set fixed text image CFG to 7.5 and camera CFG to 1.0.

C. Depth Predictor

We choose the metric depth version of Depth Anything V2 [81] as the metric depth predictor. Compared to their basic versions, the authors fine-tune the pre-trained encoder on synthetic datasets for indoor and outdoor metric depth estimation. The indoor model is capable of monocular metric depth estimation within a maximum depth of 20m. We choose Large as the model size, which has 335.3M parameters, and the indoor version. The scene scale of our model is aligned to the metric depth space of Depth Anything V2 Large Indoor, *i.e.* absolute scene scale.

D. Noise Shaping and Parameter Sensitivity

The noise shaping mechanism is straightforward. It overlays reference video features on model predictions at early diffusion steps. Fewer times of noise shaping means less prior imposed on layout, preserving more dynamics, and vice versa. Tab. 4 shows that effective control is typically achieved by selecting a threshold like $t_{\text{NS}} = 900$, similar to CFG. In practice, users only need to easily change between $\{800, 900, 1000\}$. To enhance stability, we filter outliers of depth prediction on edge regions, as shown in Fig. 11.

Noise Shaping Threshold	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	I2V Subject	I2V Background	Camera Motion
/	90.99	96.23	97.36	**46.75**	58.37	62.91	94.73	93.44	85.85
900	93.96	97.58	97.66	<u>35.77</u>	59.79	63.08	96.14	95.27	93.32
800	<u>95.02</u>	<u>97.91</u>	<u>97.79</u>	26.42	<u>60.07</u>	<u>63.33</u>	<u>96.53</u>	**95.73**	<u>95.15</u>
600	**95.10**	**98.02**	**97.80**	27.64	**60.21**	**63.85**	**96.55**	<u>95.72</u>	**96.72**

Table 4. **Evaluation results of noise shaping threshold t_{NS}.** For noise level $t \in [0, 1000]$, the threshold t_{NS} denotes that we apply noise shaping only when $t > t_{\text{NS}}$ in early denoising process.

Model	Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	I2V Subject	I2V Background	Camera Motion
CogVideoX 1.5 [83]	91.80	94.66	97.07	40.98	62.29	70.21	96.46	95.50	39.71
w. RealCam-I2V (Ours)	**97.81**	**98.41**	**99.33**	**44.31**	**64.31**	**70.70**	**99.03**	**99.40**	**91.35**

Table 5. **VBench-I2V results of RealCam-I2V trained on RealCam-Vid [98]**, with scene dynamics and large camera movements (near 360°). The improvement in metrics can be attributed to the additional information from diverse camera traces in dynamic scenes. Noise shaping significantly improves the consistency and quality.

E. Robust Analysis of Depth Estimation, SfM, Base Model and Resolution

Our framework demonstrates robustness across variations in depth predictors, SfM, base models, and high resolution. The primary goal of our depth alignment is to establish a *unified metric space*. This ensures that even if absolute depth values from predictors have errors, these errors are consistently propagated during alignment for both training and inference data. Consequently, there is no gap between training and inference despite inevitable errors from depth prediction and scale alignment. The method can function even with less accurate metric depth predictors, though better predictors naturally enhance performance of our framework.

To further address this, we reproduce our method on RealCam-Vid [98], a high-resolution video dataset with dynamic scenes and metric-scale cameras. We've successfully used both Depth Anything V2 [81] (metric indoor, 20m) and, in RealCam-Vid, Metric 3D v2 [28] (200m), showing strong adaptability to a different depth predictor, underscoring its robustness to depth estimation errors. Our method works with both UNet-based (DynamiCrafter) and DiT-based (CogVideoX 1.5 [83] in Tab. 5) backbones with higher resolution. We also explored different SfM tools (e.g., GLOMAP [48] and MonST3R [90]) and found consistent performance.

F. Real-time User Interface

Fig. 5 in main paper illustrates our shift from slow, multi-round generation to an efficient one-round workflow, significantly enhancing usability. Users intuitively design camera paths by dragging within the reconstructed 3D scene, which is more direct than 2D or numerical input. Users receive real-time preview feedback, rendered in parallel and streamed for immediate visualization, which is *optional*, catering to different preferences. Our interactive preview is designed for nearly real-time feedback. Monocular depth estimation is efficient. For preview videos (e.g., 8 sampled keyframes), each frame is rendered *independently* using modern engines (e.g., Open3D [100]) in a *parallel* manner. This allows streaming playback, achieving real-time interaction. Further engineering optimization ensures a fluid user experience.

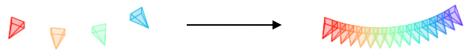

Figure 12. **Camera Trajectory Interpolation.** We interpolate camera keyframes given by user to dense trajectories.

G. Camera Keyframe Interpolation

In real-world applications, user-provided camera trajectories often consist of a limited number of keyframes (*e.g.*, 4 keyframes). To ensure smooth and continuous motion across the trajectory while adhering to the user's input, we perform linear interpolation in SE(3) space to expand the trajectory to a higher number of frames (*e.g.*, 16 interpolated frames), as shown in Fig. 12. This step ensures that our model generates consistent and visually coherent videos without compromising the accuracy of user-defined camera movements.