MotionAgent: Fine-grained Controllable Video Generation via Motion Field Agent Supplementary Materials

Xinyao Liao[1,2] Xianfang Zeng[2] Liao Wang[2] Gang Yu[2] Guosheng Lin[1] Chi Zhang[3]
[1]Nanyang Technological University [2]StepFun [3]Westlake University

1. Object Trajectory Plotting Module

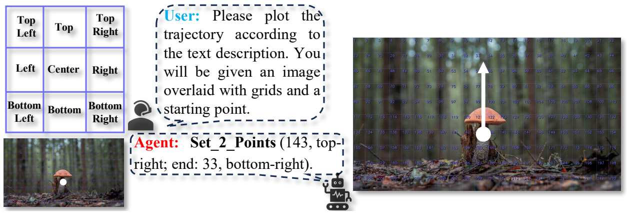
1.1. Details of Trajectory Plotting

Inspired by AppAgent [9], we replace direct trajectory generation with grid selection based on grid numbers. Specifically, we divide the given image into $N \times M$ grids, breaking it down into small square areas. Each area is labeled with an integer in the top-left corner and subdivided into nine subareas. Based on the previous step, we identify the starting point of the trajectory using the detection result and plot this starting point on the image, represented by a circle. Then, we provide the image overlaid with the grids and starting point as input to the agent.

We define the following functions: *Set_*_Points (start: int, string; mid_*: int, string; end: int, string)* for the agent. Here, * is an integer ranging from $1 - 4$, used to achieve varying lengths and curvature in trajectory plotting. The parameters *start*, *mid_**, and *end* include an integer label assigned to the grid area and a string representing the exact location within the grid area. The string can take one of the following nine values: center, top-left, top, top-right, left, right, bottom-left, bottom, and bottom-right.

A simple use case is *Set_2_Points (start: 143, top-right; end: 33, bottom-right)*, which sets the starting point of the trajectory at the top-right of grid area 143 and the endpoint at the bottom-right of grid area 33. As illustrated in Figure 1, this function represents a simple linear trajectory.

Once we obtain the complete object trajectory, we use interpolation to determine the position in the trajectory for each frame. Then, these interpolated positions are input into the subsequent model to calculate dense optical flow.



User: Please plot the trajectory according to the text description. You will be given an image overlaid with grids and a starting point.

Agent: **Set_2_Points** (143, top-right; end: 33, bottom-right).

Figure 1. Details of trajectory plotting: The grid divides the image into small square areas. Each area is labeled with an integer in the top-left corner and is further subdivided into nine subareas.

Offset Generation	Grid Selection	Object Movement Q&A	Dynamic Degree
✓		29.28	7.63
	✓	45.69	32.11

Table 1. Ablation study of trajectory plotting module (all values are in percentage).

1.2. Ablation Study

Here, we conduct an ablation study on different approaches to trajectory plotting. We use direct offset generation as the baseline module instead of grid selection. Specifically, we provide the agent with the starting point location based on the detection results and ask the agent to directly generate offsets from the starting point to define the object trajectory.

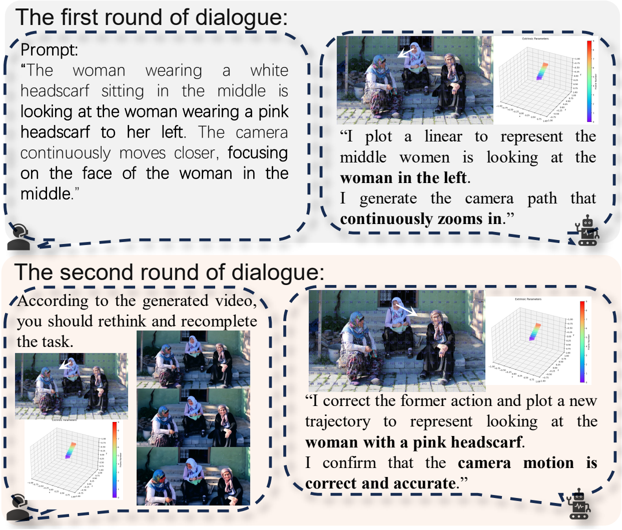
As shown in Table 1, the grid selection approach shows better evaluation results in both metrics. The grid selection approach gives the agent an overall understanding of the image layout and is easier for the agent to use than direct offset generation. As for dynamic degree metrics, the offset generation approach cannot output a suitable trajectory length, which may lead to a lower dynamic degree.

2. Rethinking Module

We report the results on VBench [1] after applying the rethinking step. As shown in Table 2, the rethinking mechanism achieves higher scores across most metrics. It corrects errors in camera motion generated by earlier stages, leading to notable improvements in the Video-Text Camera Motion metric. Visual quality is also a key consideration in the rethinking step. By refining object movement trajectories and adjusting the range of camera motion, this step effectively reduces artifacts in the generated videos, as reflected in the improved video quality metrics. To reduce such artifacts, the rethinking step tends to shorten object movement trajectories, which may explain the slight decrease observed in the Dynamic Degree metric.

Method	Video-Text Camera Motion	Video-Image Subject Consistency	Video-Image Background Consistency	Subject Consistency	Background Consistency	Motion Smoothness	Aesthetic Quality	Dynamic Degree
MotionAgent	81.91	98.06	98.00	96.10	96.76	98.93	64.48	**16.67**
MotionAgent (Rethinking)	**87.02**	**98.22**	**98.12**	**96.44**	**96.85**	**99.08**	**64.69**	15.38

Table 2. Evaluation results of general I2V generation on VBench [1] (all values are in percentage). The best result is indicated in **bold**.



The first round of dialogue:

Prompt:
"The woman wearing a white headscarf sitting in the middle is looking at the woman wearing a pink headscarf to her left. The camera continuously moves closer, focusing on the face of the woman in the middle."

"I plot a linear to represent the middle women is looking at the **woman in the left**.
I generate the camera path that **continuously zooms in**."

The second round of dialogue:

According to the generated video, you should rethink and recomplete the task.

"I correct the former action and plot a new trajectory to represent looking at the **woman with a pink headscarf**.
I confirm that the **camera motion is correct and accurate**."

Figure 2. Example of rethinking step.

Figure 2 shows an example of the rethinking step. The agent corrects the previous error response and confirms the correct action that it made the last time, which facilitates the generation of better video results.

3. Complex and Ambiguous Prompts.

We rewrite the motion prompts in our designed sub-VBench using more complex or ambiguous descriptions to evaluate generalization ability and robustness. In Table 3, we report comparative results on these rewritten prompts. For complex prompt inputs, our method maintains comparable performance on object movement metrics and shows a slight decline in camera motion compared to the results in Table 2 (Main Body). For ambiguous prompt inputs, performance degradation is observed in both metrics relative to the simple prompts in Table 2 (Main Body). Nevertheless, MotionAgent still achieves competitive results and outperforms other I2V generation methods in achieving semantic alignment between motion prompts and the generated videos.

4. Training Data Preparation

To eliminate the domain gap between the unified and real optical flow maps, we propose fine-tuning the optical flow adapter module, which maintains the generation capabilities of the base I2V diffusion model. For each video used for training, we first utilize a binary segmentation model, BiRefNet [10], to decompose the foreground and background. We then remove the dynamic foreground based on the binary segmentation mask. Next, we adopt an SLAM method, DROID-SLAM [3], to compute the camera extrinsics $\hat{E}$ from the masked video and the Metric3D [7] to estimate the depth map $\hat{D}$ for every frame. Additionally, for the original video, we use an optical flow model, Unimatch [5], to estimate the real optical flow $\hat{F}$.

Next, we explain how to estimate the optical flow caused by object movement based on the camera extrinsics $\hat{E}$, depth map $\hat{D}$ and real optical flow $\hat{F}$. We define I^0 as the pixel position in the first frame. According to the predicted real optical flow $\hat{F}$, we compute the corresponding pixel position in the following frames, which can be formulated as,

$$I^1 = I^0 + \hat{F}. \tag{1}$$

Then, we reproject the pixel position in the following frames back to the image coordinate systems of the first frame according to the depth map $\hat{D}$ and the predicted camera extrinsics $\hat{E}$. This can be computed by,

$$I^1_{obj} = \Pi(\hat{E}^{-1}\Pi^{-1}(I^1)), \tag{2}$$

where Π and Π^{-1} are the projection and unprojection operations, respectively. We assume that the camera coordinate of the first frame serves as the world coordinate. I^1_{obj} indicates the corresponding pixel position of the following frames in the image coordinate systems of the first frame, which contains only object movement. Finally, we compute the optical flow caused by object movement,

$$\hat{F}_{obj} = I^1_{obj} - I^0. \tag{3}$$

We perform sampling [8] on these optical flow maps of object movement $\hat{F}_{obj}$ to obtain sparse object trajectories. Subsequently, we reuse the proposed analytical composition method to calculate the unified optical flow maps F, which are utilized as input to fine-tune the optical flow adapter. Additionally, we calculate the error between the unified optical flow maps F and the real optical flow $\hat{F}$. If the error exceeds a threshold, we replace the unified optical flow maps F with the real optical flow $\hat{F}$ for training.

Method	Complex Prompts			Ambiguous Prompts		
	Object Movement Q&A	Camera Motion	Total Scores	Object Movement Q&A	Camera Motion	Total Scores
CogVideoX [6]	24.01	16.12	19.24	20.71	11.00	14.48
Pyramid Flow [2]	26.39	5.95	14.03	17.45	4.49	9.61
MotionAgent	45.78	73.84	62.75	37.07	66.28	54.74
MotionAgent (Rethinking)	**48.19**	**79.32**	**67.02**	**42.28**	**69.37**	**58.67**

Table 3. Results of complex and ambiguous controllable I2V generation.

5. Metrics Details

In the comparison experiments of the general I2V generation task, we adopt the same evaluation metrics introduced by VBench [1]. In the comparison of controllable I2V generation, we report Object Movement Q&A, Complex Camera Motion, and Overall Score.

I2V Score reports the overall score of I2V generation metrics. **Video-Text Camera Motion** assesses the consistency between camera motion and the input text, such as zoom in/out. **Video-Image Subject Consistency** assesses whether the appearance of the subject remains consistent throughout the entire video compared to the input image. **Video-Image Background Consistency** evaluates the temporal consistency of background scenes with the input image. **Subject Consistency** assesses whether the subject's appearance remains consistent throughout the entire video. **Background Consistency** evaluates the temporal consistency of the background scenes across frames. **Motion Smoothness** evaluates whether the motion in the generated video is smooth and follows the physical laws of the real world. **Aesthetic Quality** evaluates the artistic and aesthetic value perceived by humans towards each video frame. **Dynamic Degree** evaluates the level of dynamics generated by each model. **Object Movement Q&A** assesses the consistency between the text description and object movement in the video. **Complex Camera Motion** evaluates the consistency between complex camera movements in the generated video and the input text description. **Total Score** is the overall metric of controllable I2V generation.

6. Dynamic Degree

Our method performs a relative lower dynamic degree in the original prompt provided by Vbench [1], we claim that the decrease is due to the precise control we implement over the generated video. The objects mentioned in the text move accurately, while those not mentioned remain as static as possible, which leads to a lower dynamic degree score. In Figure 3, we show the result of the comparison with Dynamicrafter [4] and demonstrate that our method follows the motion information in the text prompt precisely.

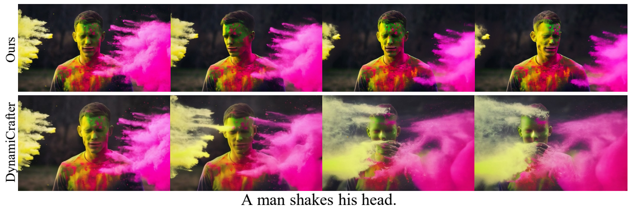


Figure 3. Dynamic degree comparison with DynamiCrafter [4].

7. Qualitative Results

7.1. Fine-grained Controllable Video Results

In Figure 4, we show more fine-grained controllable video results generated by our method.

7.2. Rethinking Results

As shown in Figure 5, the rethinking step corrects the inaccurate object movement (subfigure a, complex prompt) and enhances the quality of the generated video (subfigure b, ambiguous prompt).

7.3. Visualization of Intermediate Representations

As illustrated in Figure 5, we show the intermediate representations generated by the motion field agent.

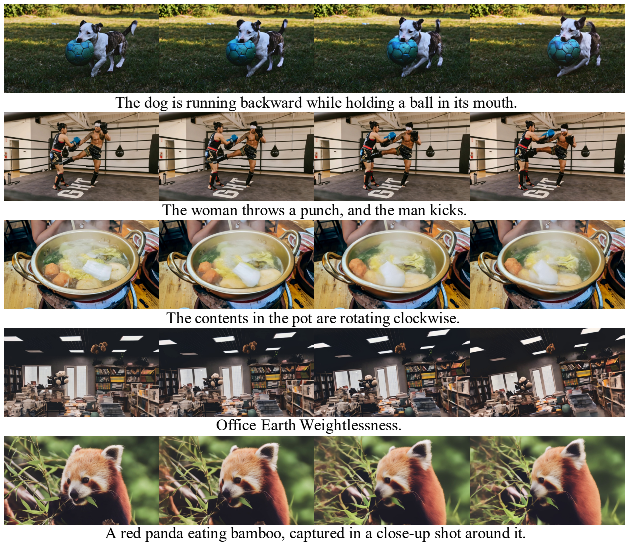
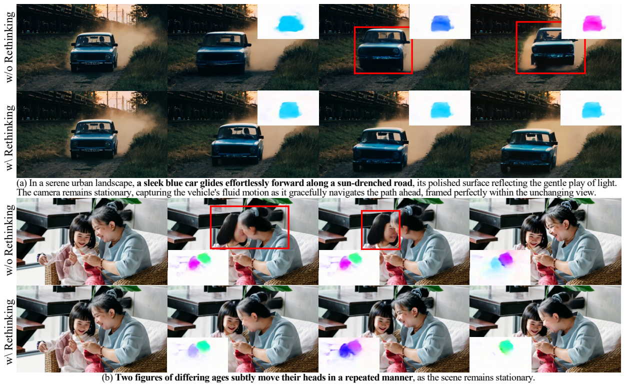

The dog is running backward while holding a ball in its mouth.
The woman throws a punch, and the man kicks.
The contents in the pot are rotating clockwise.
Office Earth Weightlessness.
A red panda eating bamboo, captured in a close-up shot around it.

Figure 4. More fine-grained controllable video results generated by our method.



(a) In a serene urban landscape, **a sleek blue car glides effortlessly forward along a sun-drenched road**, its polished surface reflecting the gentle play of light. The camera remains stationary, capturing the vehicle's fluid motion as it gracefully navigates the path ahead, framed perfectly within the unchanging view.

(b) **Two figures of differing ages subtly move their heads in a repeated manner**, as the scene remains stationary.

Figure 5. Visualization results for rethinking and optical flow.

7.4. Qualitative Comparison Results

In Figure 6, we show more results compared to DynamiCrafter [4], CogVideoX [6] and Pyramid Flow [2].

8. User Study Interface

The designed user study interface is shown in Figure 7. For each question, we randomly shuffle four videos generated by our method and the other three methods. We then ask participants to rank the videos from highest to lowest twice based on specific requirements. After the user study, we calculate the mean ranking for each method across different evaluation dimensions.

9. Prompts

In Table 4, 5, 6, we present some majority prompts used in our method for object trajectory plotting, camera extrinsics generation, and rethinking.

References

[1] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 1, 2, 3

[2] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024. 3, 4

[3] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 2

[4] 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 *European Conference on Computer Vision*, pages 399–417. Springer, 2025. 3, 4

[5] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

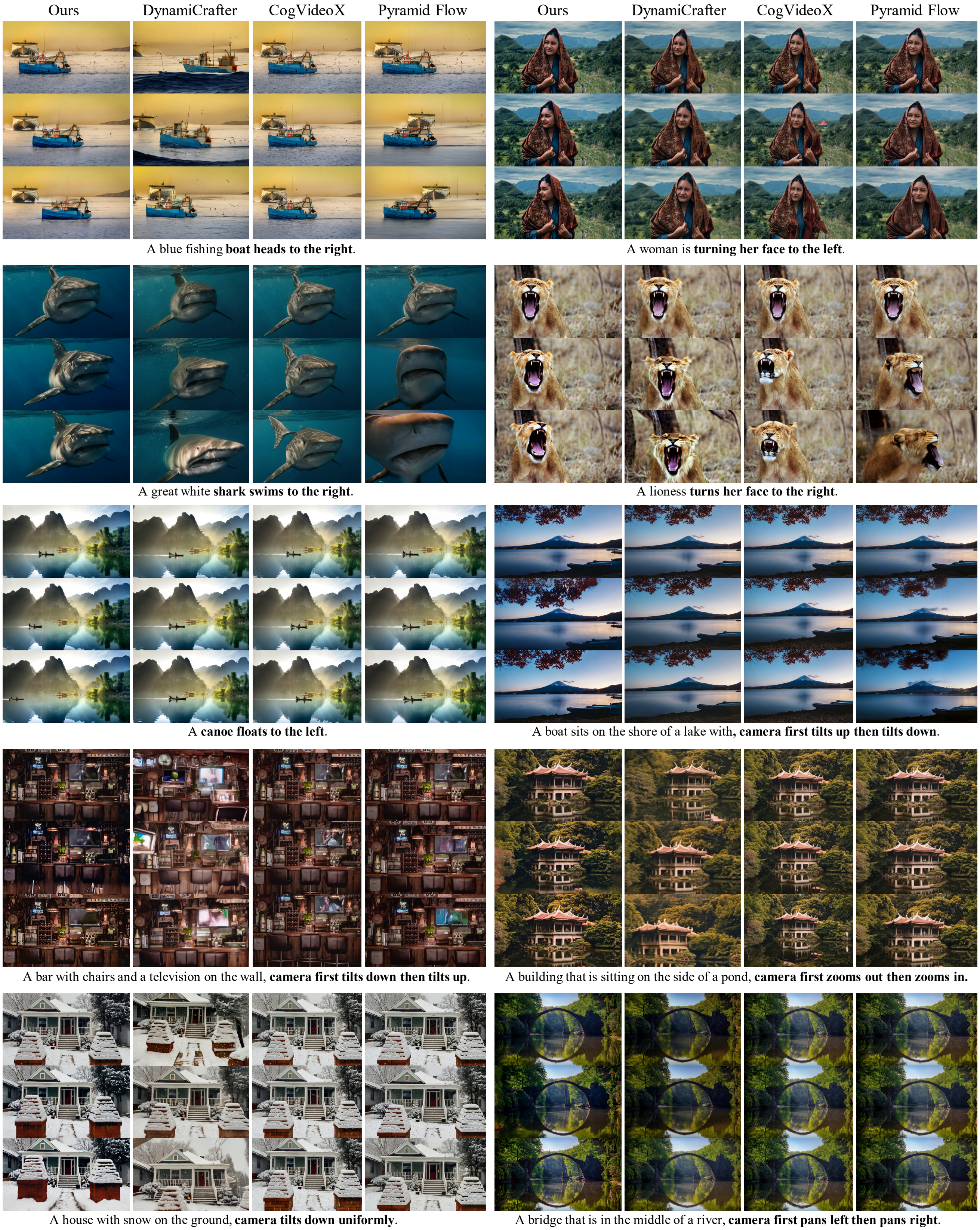
[6] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3, 4

[7] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9043–9053, 2023. 2

[8] Xiaohang Zhan, Xingang Pan, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised learning via conditional motion propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1881–1889, 2019. 2

[9] Chi Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*, 2023. 1

[10] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 3:9150038, 2024. 2



A blue fishing **boat heads to the right**.

A woman is **turning her face to the left**.

A great white **shark swims to the right**.

A lioness **turns her face to the right**.

A **canoe floats to the left**.

A boat sits on the shore of a lake with**, camera first tilts up then tilts down**.

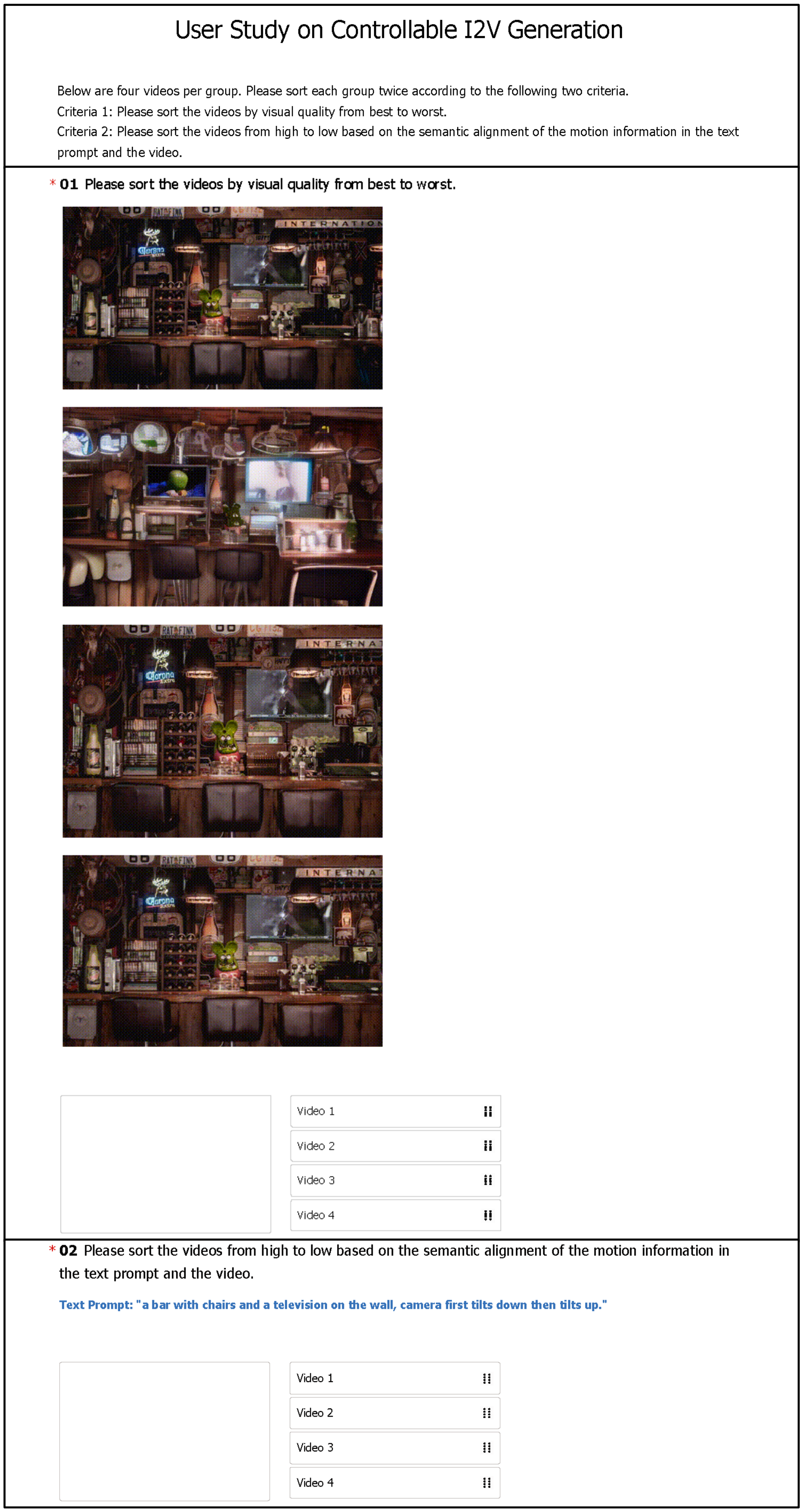
A bar with chairs and a television on the wall, **camera first tilts down then tilts up**.

A building that is sitting on the side of a pond, **camera first zooms out then zooms in.**

A house with snow on the ground, **camera tilts down uniformly**.

A bridge that is in the middle of a river, **camera first pans left then pans right**.

Figure 6. More comparison results of controllable I2V generation on our benchmark. The motion described in the text is in **bold**.



User Study on Controllable I2V Generation

Below are four videos per group. Please sort each group twice according to the following two criteria.
Criteria 1: Please sort the videos by visual quality from best to worst.
Criteria 2: Please sort the videos from high to low based on the semantic alignment of the motion information in the text prompt and the video.

* **01** Please sort the videos by visual quality from best to worst.

Video 1
Video 2
Video 3
Video 4

* **02** Please sort the videos from high to low based on the semantic alignment of the motion information in the text prompt and the video.

Text Prompt: "a bar with chairs and a television on the wall, camera first tilts down then tilts up."

Video 1
Video 2
Video 3
Video 4

Figure 7. User study interface. Each participant is required to evaluate 30 groups of videos and respond to two corresponding sub-questions for each group. Only one group of videos and two sub-questions are shown here due to the page limit.

Prompt Template: Object Trajectory Plotting

You are an agent trained to plot a trajectory on an image based on a text and a starting point. You will be given an image overlaid with a grid and a starting point. The grid divides the image into small square areas. Each area is labeled with an integer in the top-left corner. The starting point of the trajectory is represented by a circle.

You can call the following functions to plot a trajectory:
- ......
- **Set_3_Points(start_area: int, start_subarea: str, mid_area: int, mid_subarea: str, end_area: int, end_subarea: str):**
  This function is used to set a starting point, a mid-point and an end point of a trajectory, which represents a complex trajectory. start_area is the integer label assigned to the grid area, marking the trajectory's starting location. start_subarea is a string representing the exact location to begin the trajectory within the grid area......, The three subareas' parameters can take one of the nine values: center, top-left, top, top-right, left, right, bottom-left, bottom, and bottom-right.
- ......

The task you need to complete is to plot a trajectory to describe: `<task_description>`. The location of the trajectory starting point is: `<start_point_location>`. Now, given the following labeled image, you need to think and call the function needed to proceed with the task:
- First, find the location of the object according to the task description and set the same starting point as the given one.
- Next, select N midpoints to extend the trajectory.
- Finally, select an end point to complete the trajectory.

Your output should include four parts in the given format:
- **Observation:** Describe what you observe in the image.
- **Thought:** To complete the given task, what is the step you should take.
- **Action:** The function call with the correct parameters to proceed with the task.
- **Summary:** Summarize your actions in one or two sentences.

Table 4. Prompt template for Object Trajectory Plotting.

Prompt Template: Camera Extrinsics Generation

You are an agent trained to generate a camera motion based on a text and an image. please note that the world coordinate is opencv's one, which is x-axis rightwards (camera pans right), y-axis downwards (camera tilts down), and z-axis frontwards (camera zooms in).

You can call the following functions to generate a camera motion:

- ......
- **Set_Camera_Motion(x_translation: float, y_translation: float, z_translation: float, x_rotation: int, y_rotation: int, z_rotation: int, motion_type: str):**
  This function sets a simple camera motion, such as pan down, that is represented by the shifting distance and rotation degrees of the camera optical center on the x-axis, y-axis, and z-axis. x_translation is a floating point ranged from (-1.00, 1.00), which represents the shift distance in the x axis......, x_rotation is a integer ranged from (0, 360), which represents the degrees rotated alone the x axis......, motion_speed is a string representing the camera motion type, and the parameters can take one of the three values: uniform, decrement, increment.
- ......

The task you need to complete is to generate a camera motion to describe: `<task_description>`. Now, given the following image, you need to think and call the function needed to proceed with the task:

- First, imagine that the given image is shot at the initial location of the camera.
- Then, analyze the text description and the image content to determine the direction and distance of the following camera motion.
- Finally, call the function with the correct parameters to generate the camera motion.

Your output should include four parts in the given format:

- **Observation:** Describe what you observe in the image.
- **Thought:** To complete the given task, what is the step you should take.
- **Action:** The function call with the correct parameters to proceed with the task.
- **Summary:** Summarize your actions in one or two sentences.

Table 5. Prompt template for Camera Extrinsics Generation.

Prompt Template: Rethinking

You are an agent that is trained to rethink and recomplete a specific task about video generation. I will provide you some frames of the generated video, which is generated based on the former action you made. Additionally, I will describe the task that you should recomplete and give the action you made at the last time.

- According to the task description and the generated video, you should first analyze the former action.
- Then, you should correct the error in the former action.
- Finally, you should recomplete the task and take the right action at this time.
- If you think the action you made last time is correct, you can recomplete the task with the same action.

The generated video is: ......

The task you should recomplete is: ......

The action you made last time is: ......

Table 6. Prompt template for Rethinking.