



# WonderZoom: Multi-Scale 3D World Generation

## Supplementary Material

### A. Algorithm

We provide an algorithm of WonderZoom in Alg. 1

### B. Additional Results

We provide additional visual results in Figures 9, 10, 11 and 12 to show that WonderZoom significantly outperforms other baselines in terms of visual quality.

### C. Failure Cases

As shown in Fig. 13, when zooming repeatedly into the cluster of branches, the scene eventually collapses into pure texture patterns with no remaining semantic cues (e.g., individual branches or leaves). Since WonderZoom relies on the semantics of the current-scale image to infer what should appear at the next scale, such texture-only regions become under-constrained, making further refinement in more new scales unreliable, and finally fail to generate a multi-scale 3D world.

This failure does not occur when recognizable structure is still present, but represents an inherent limitation when the input region no longer contains semantic information.

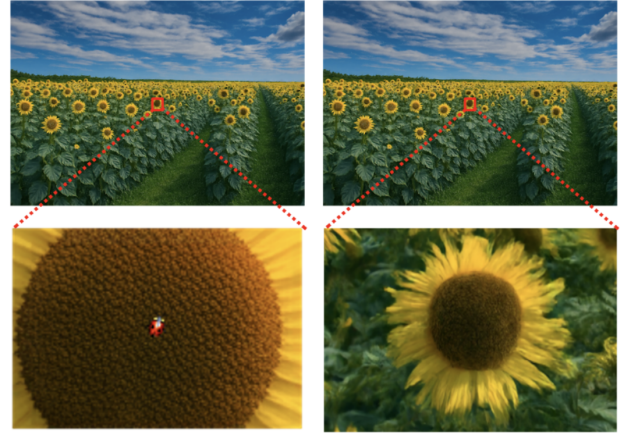
### D. Additional Details

**Additional implementation details.** All images are processed at a resolution of  $720 \times 1088$ . We use GPT-4V as our VLM for semantic context extraction and editing prompt generation. The initial camera focal length is set to  $f_x = f_y = 1024$ , with progressive zoom-in operations increasing the focal length for finer scales, typically we multiply the current focal length by 8 for a new scale. We use INR-Harmonization [6] after image editing for improved shading consistency.

**Human study details.** We use Prolific to recruit participants for our human preference evaluation. For each comparison, we collect responses from around 200 participants worldwide. The survey is implemented using Google Forms, and all responses are fully anonymized for both the participants and the authors. Each question presents two zoom-in sequences of the same scene generated by two different methods. Participants are shown the images in a left-right layout: each side contains (1) a global view of the scene and (2) a zoomed-in view of the same region, indicated by a red bounding box and connecting lines. The left-right order of methods is randomized for every participant and every question. Participants are instructed to carefully compare the two sides and make a two-alternative forced choice (2AFC).

Your task:

Compare the left side and right side, and answer the questions below.



Prompt: A ladybug is on the sunflower

	left side	right side
Which one looks more like the camera is getting closer?	<input type="radio"/>	<input type="radio"/>
Which one looks better to your eyes?	<input type="radio"/>	<input type="radio"/>
Which one matches the text the best?	<input type="radio"/>	<input type="radio"/>

Figure 8. An example of our user study.

For each comparison, we ask three questions: (i) “Which one looks like the camera is moving closer?” (ii) “Which one looks better to your eyes?” and (iii) “Which one fits the prompt better?” We compare our method with four baselines across six scenes, this yields 24 comparison pairs and 72 questions in total. Each participant answers all 72 questions. A screenshot of the survey interface is provided in Figure 8.

**Algorithm 1** Multi-Scale 3D World Generation Control Loop

---

**Input:** Initial image  $\mathbf{I}_0$ , initial camera  $\mathbf{C}_0 \in \mathbb{R}^{4 \times 4}$   
**Output:** Multi-scale scene hierarchy  $\{\mathcal{E}_0, \mathcal{E}_1, \dots, \mathcal{E}_n\}$   
**Runtime output:** Real-time rendered observation  $\mathbf{O}_{\text{render}}$   
**Runtime user control:** Camera viewpoint  $\mathbf{C}_{\text{render}}$ , zoom region  $\mathbf{C}_{i+1}$ , (optional) edit prompt  $\mathcal{U}_{i+1}$

---

```

1: Initialize:  $\mathcal{E}_0 \leftarrow \text{ReconstructScene}(\mathbf{I}_0, \mathbf{C}_0)$  ▷ Initial 3D scene from input image
2:  $\mathbf{C}_{\text{render}} \leftarrow \mathbf{C}_0$  ▷ Initialize rendering camera
3:  $i \leftarrow 0$  ▷ Current scale index

4: Thread 1: Real-time Scale-Adaptive Rendering ▷ Continuous rendering loop
5: while true do
6:    $s^{\text{render}} \leftarrow d^{\text{render}} / \sqrt{f_x^{\text{render}} f_y^{\text{render}}}$  ▷ Compute rendering scale
7:    $\mathbf{O}_{\text{render}} \leftarrow \text{RenderWithOpacityModulation}(\bigcup_{k=0}^i \mathcal{E}_k, \mathbf{C}_{\text{render}})$  ▷ Sec. 3.1
8:    $\mathbf{C}_{\text{render}} \leftarrow \text{UserCameraControl}()$  ▷ Interactive camera update
9: end while

10: Thread 2: Progressive Detail Synthesis ▷ Triggered by user zooming into region of interest with prompt  $\mathcal{U}_{i+1}$  at camera  $\mathbf{C}_{i+1}$ 
11: // Stage 1: New Scale Image Synthesis
12:  $\mathbf{O}_{i+1} \leftarrow \text{Render}(\mathcal{E}_i, \mathbf{C}_{i+1})$  ▷ Coarse observation at zoomed view
13:  $\mathcal{S} \leftarrow \text{VLM}(\text{Render}(\mathcal{E}_i, \mathbf{C}_i))$  ▷ Extract semantic context
14:  $\mathbf{I}'_{i+1} \leftarrow \text{SuperResolution}(\mathbf{O}_{i+1}, \mathcal{S})$  ▷ Extreme super-resolution
15: if  $\mathcal{U}_{i+1} \neq \emptyset$  then
16:    $\mathbf{I}_{i+1} \leftarrow \text{ControlledEdit}(\mathbf{I}'_{i+1}, \mathcal{U}_{i+1})$  ▷ Insert user-specified content
17: else
18:    $\mathbf{I}_{i+1} \leftarrow \mathbf{I}'_{i+1}$ 
19: end if
20: // Stage 2: Scale-Consistent Depth Registration
21:  $\mathbf{D}_{i+1}^{\text{target}} \leftarrow \text{RenderDepth}(\mathcal{E}_i, \mathbf{C}_{i+1})$  ▷ Target depth from coarse scale
22:  $\mathbf{D}_{i+1} \leftarrow \text{DepthRegistration}(\mathbf{I}_{i+1}, \mathbf{D}_{i+1}^{\text{target}})$  ▷ Fine-tune depth estimator
23: // Stage 3: Scale-Adaptive Surfel Generation
24:  $\mathcal{E}_{i+1}^{\text{partial}} \leftarrow \text{InitializeSurfels}(\mathbf{I}_{i+1}, \mathbf{D}_{i+1}, \mathbf{C}_{i+1})$ 
25: ▷ Create surfels with  $s^{\text{native}} = d^{\text{native}} / \sqrt{f_x^{\text{native}} f_y^{\text{native}}}$ 
26: // Stage 4: Auxiliary View Synthesis
27:  $\{\mathbf{C}_{i+1}^k\}_{k=1}^K \leftarrow \text{SampleNeighboringViews}(\mathbf{C}_{i+1})$ 
28:  $\{\mathbf{I}_{i+1}^k, \mathbf{D}_{i+1}^k\} \leftarrow \text{AuxiliaryViewSynthesis}(\mathcal{E}_{i+1}^{\text{partial}}, \{\mathbf{C}_{i+1}^k\})$ 
29: // Stage 5: Optimization
30:  $\mathcal{E}_{i+1} \leftarrow \text{OptimizeSurfels}(\mathcal{E}_{i+1}^{\text{partial}}, \{\mathbf{I}_{i+1}, \mathbf{I}_{i+1}^1, \dots, \mathbf{I}_{i+1}^K\})$ 
31: ▷ Optimize  $\{\mathbf{q}, \mathbf{s}, o\}$  with  $\mathcal{L} = 0.8L_1 + 0.2L_{\text{D-SSIM}}$ 
32:  $i \leftarrow i + 1$  ▷ Increment scale index

```

---

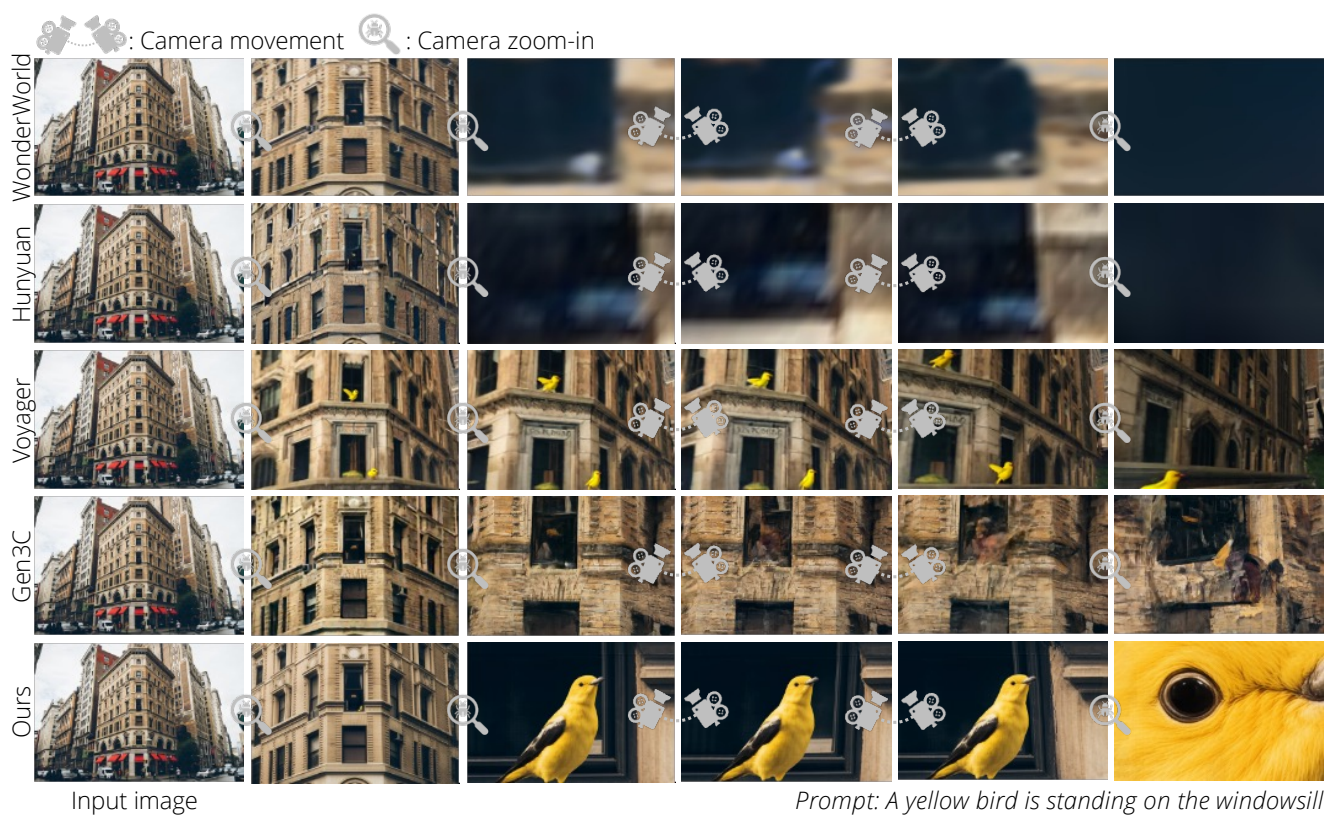


Figure 9. Visual comparison of multi-scale 3D world generation results.





Figure 10. Visual comparison of multi-scale 3D world generation results.



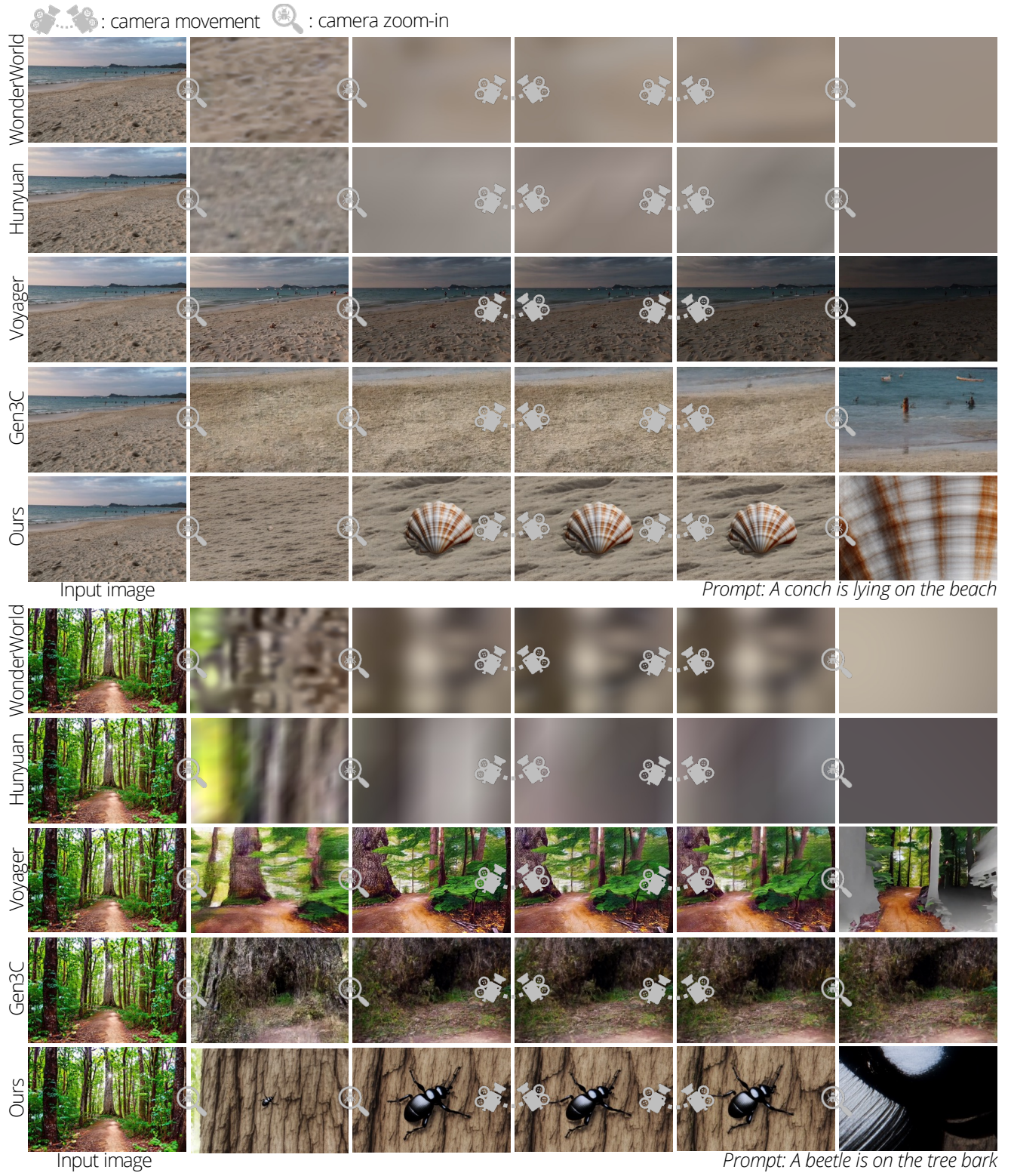
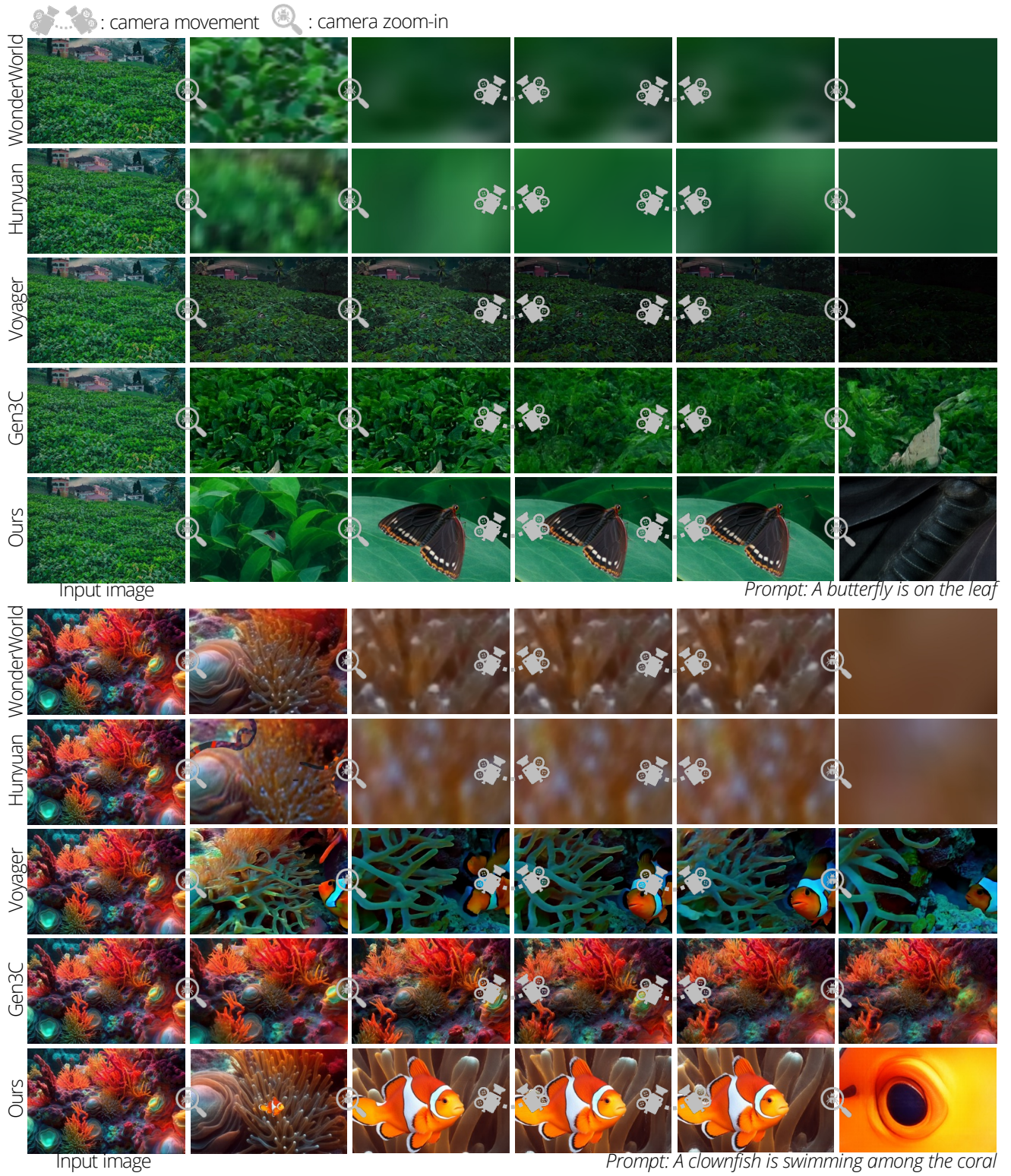


Figure 11. Visual comparison of multi-scale 3D world generation results.





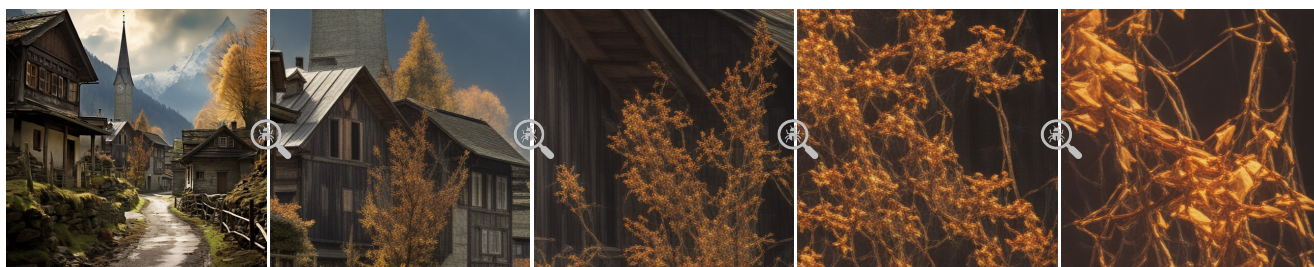


Figure 13. A failure case of WonderZoom. When zooming too deeply into the tree region, the view collapses into texture-like patterns instead of meaningful branch structures.