DynamicTree: Interactive Real Tree Animation via Sparse Voxel Spectrum

Supplementary Material

1. Dataset

As discussed in the paper, we implement the method of [2] as a plugin in Blender. However, generating a large-scale 4D tree dataset using this method remains challenging, as it involves numerous parameters and random sampling often produces invalid or unrealistic 4D trees. To ensure data quality and consistency, we adopt a three-stage pipeline to construct our dataset:

1. **Parameter Tuning**: Trees are controlled by many shape parameters. Fully random sampling over all of them tends to generate irregular or unrealistic trees, which can harm network training. Instead, we manually select key parameters such as branch count, height, branching angle, leaf count, etc., for stochastic variation. Other parameters are kept within small perturbation ranges. This approach ensures diversity while avoiding extreme or implausible deformations.
2. **Automatic Filtering**: After generating approximately 10,000 trees using the above strategy, we observe that some samples exhibit undesirable high-frequency oscillations, such as rapid back-and-forth motion at the root or excessive shaking in small branches. To filter out these cases, we apply the Fast Fourier Transform to each motion sequence and remove samples where the high-frequency components exceed a threshold.
3. **Manual Curation**: Finally, we perform visual inspection to eliminate edge cases such as unnatural branch clustering or physically implausible motion patterns.

Through this process, we curate a final set of about 8.5k 4D trees, with selected examples visualized in Fig. 1. For each tree, we first apply the FFT to its motion and then voxelize it. The spectrum of vertices within the same voxel is averaged to produce the final sparse voxel spectrum representation.

2. Network

The sparse encoder and sparse voxel diffusion U-Net are adapted from the basic modules proposed in XCube to better fit our conditioning input and spectral output. We report key parameter settings of these two components in Table 1

Table 1. Architecture Parameters

Parameter	Sparse Encoder	Voxel Diffusion
Base channels	32	128
Depth	3	2
Channels multiple	-	[1, 2, 4, 4]
Head	-	8
Attention Resolution	-	[4,8]

3. Visualization results

Further visualizations and analytical details of the real-world 3D tree animations and interactive dynamic simulations are presented in Fig. 3 and Fig. 2. Moreover, to facilitate a comprehensive perceptual and qualitative evaluation of our method, we strongly recommend reviewing our video results on the project website.

References

[1] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7396–7405, 2020. 3

[2] Jason Weber and Joseph Penn. Creation and rendering of realistic trees. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 119–128, 1995. 1

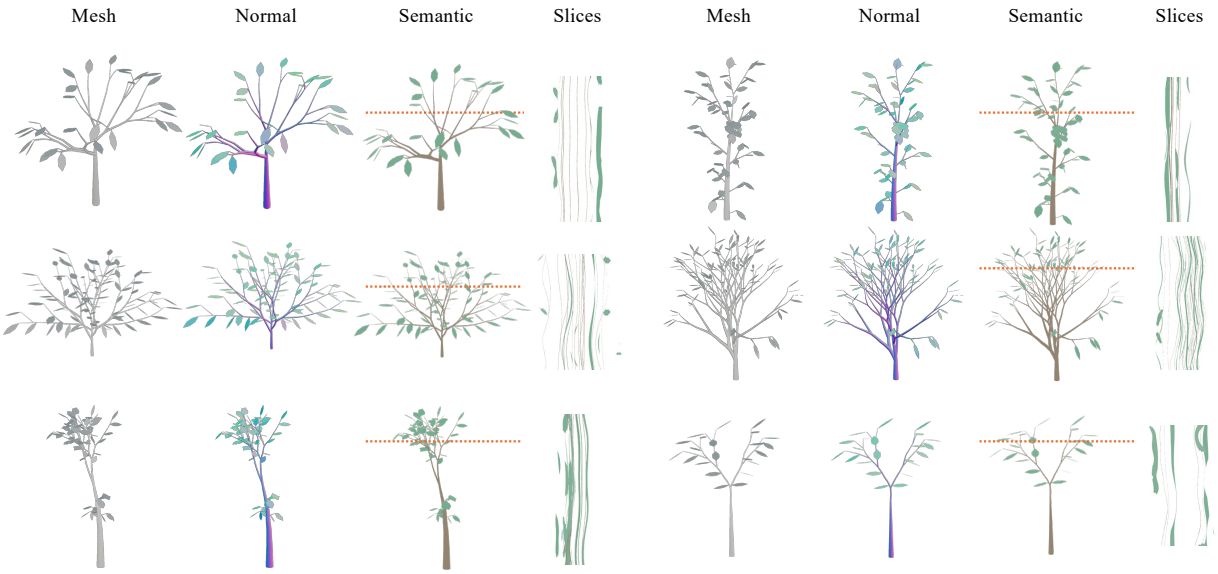

Figure 1. Examples from our synthetic dataset. To demonstrate the semantic labels, we render the leaves and trunk with two simple material settings. Users can replace these with more realistic materials for enhanced visual quality.

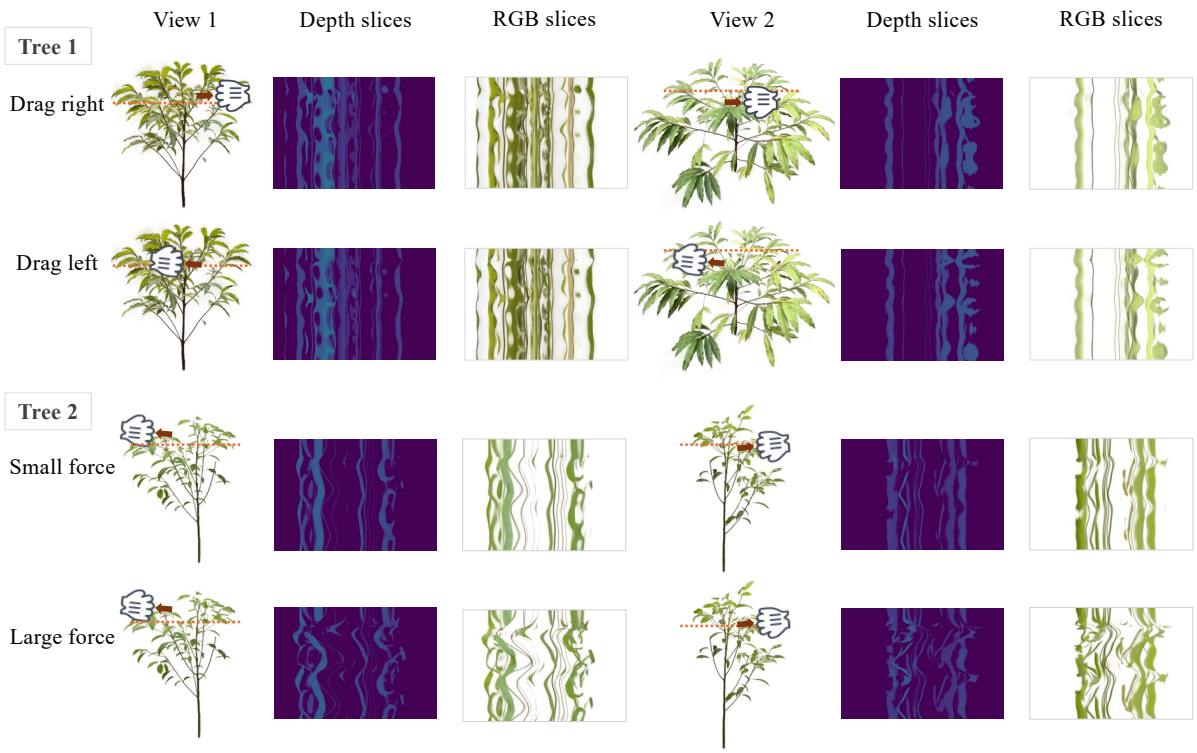

Figure 2. More results of interactive dynamic simulation. Our method can support interactive simulations involving forces with varying magnitudes and directions.

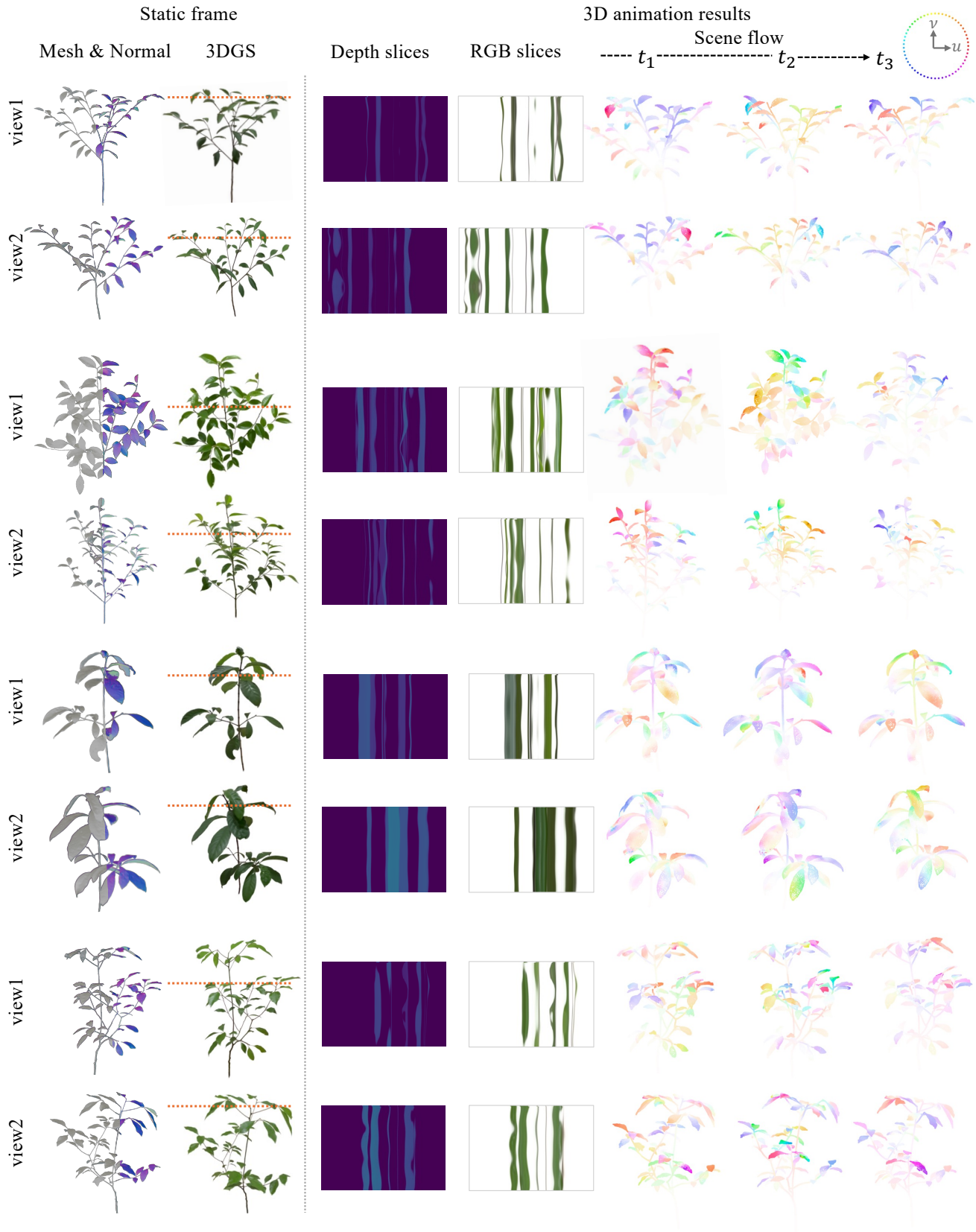

Figure 3. More results of 3D animation. For each scene, we visualize the space-time slices of depth and RGB videos from two viewpoints. We also show the scene flow of three mesh point cloud frames ($t_1 = 30, t_2 = 50, t_3 = 80$) in the generated sequence, with color coding following the strategy used in [1], where the movements (u, v) along the x and z directions are encoded using standard optical flow coloring.