Interactive Modeling of Trees using VR Devices

Zhihao Liu* [1,2], Ce Shen †[1,2], Zhi Li ‡[2,3], Tingyu Weng §[2], Oliver Deussen ¶[1,4], Zhanglin Cheng ‖[1], and Dangxiao Wang **[5,6,7]

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Institute of Automation, Chinese Academy of Sciences
[4]University of Konstanz, Germany
[5]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[6]Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University
[7]Peng Cheng Laboratory, Shenzhen

Abstract

Conventional interactive 3D tree modeling systems are generally based on 2D input devices, and it's not convenient to generate desired 3D tree shape from 2D inputs due to the complexity of 3D tree structures. In this paper, we present a system for modeling trees interactively using a 3D gesture-based VR platform. The system contains a head-mounted display (HMD) and a 6-DOF motion controller for interaction. We propose an improved procedural modeling method to generate trees faster for VR platform. Using the 6-DOF motion controller, users can manipulate tree structures by a set of 3D interactive operations, including geometric editing using 3D gestures, sketching, brushing and silhouette-guided growth. Our interactions are more flexible and convenient than using traditional 2D input devices, e.g., we allow the user to simultaneously rotate and translate parts of a tree using a 3D gesture.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Computing methodologies—Computer graphics—Shape modeling

1 Introduction

Realistic tree models play an important role in computer and VR applications. Due to their enormous structural complexity, modeling trees manually with traditional modeling software is quite inefficient and complicated. Hence, numerous specialized plant modeling approaches have been developed to automatically synthesize realistic plant models. Procedural modeling is an important form of such modeling approaches [8, 14, 15]. It assumes that trees have a recursive and repetitive architecture and adopts rule-based or self-organizing strategies for tree modeling. Although these methods can generate different species of trees, it is hard for users to control the plant structure and determine proper rules and parameters for goal-oriented modeling. To control the shape of tree effectively, many interactive tree modeling approaches have been proposed. Sketch-based methods [12, 20] infer 3D structures from 2D sketches.

*e-mail: liuzh96@outlook.com
†e-mail: ce.shen@siat.ac.cn
‡e-mail: lizhi181@mails.ucas.ac.cn
§e-mail: wengtingyu18@mails.ucas.edu.cn
¶e-mail: oliver.deussen@uni-konstanz.de
‖e-mail: zl.cheng@siat.ac.cn
**e-mail: hapticwang@buaa.edu.cn
Corresponding authors: Zhanglin Cheng, Dangxiao Wang.

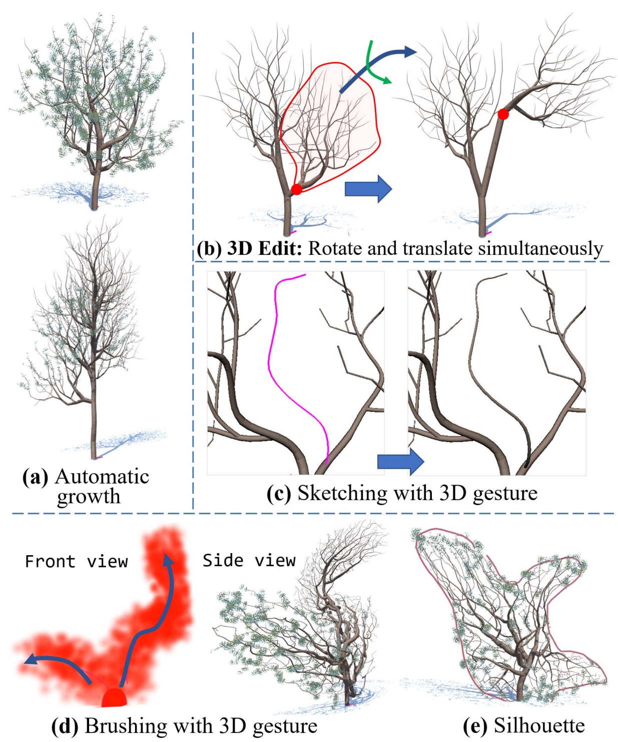

Figure 1: Modeling operations and results of our interactive system.

However, existing interactive systems are generally based on traditional 2D input devices, such as a mouse or a digital pen, which are often inconvenient for users due to the complex 3D structures of trees.

VR immerses users into a virtual environment and is able to provide flexible and continuous interactions using 3D gestures. Taking advantage of this, we developed an interactive system for modeling trees based on a VR platform. Our aim is to combine the advantages of VR devices and procedural modeling. We provide five main mechanisms to create a tree as shown in Fig. 1, these include automatic growth, rotation and translation of branches, sketching branches, brushing density fields for the foliage as well as editing the silhouette. The system contains an HMD for display and a 6-DOF motion controller for interaction (see Fig. 2). Our setup allows users to perform translations and rotations simultaneously using the

978-1-7281-4752-9/19/$31.00 ©2019 IEEE
DOI 10.1109/ICVRV47840.2019.00020

Figure 2: Interactive tree modeling system.

motion controller, users can change their position by head rotation instead of adjusting a camera using the mouse during editing. We also provide a more precise method to control the angle of rotation using the touchpad of the controller. Tree growth can be guided by a 3D-brush and the intended silhouette of the model. All parameters for modeling can be changed using the touchpad during brushing.

The contributions of our work include the following two aspects: (1) We improve the procedural modeling method using guided and weighted markers to generate trees faster for VR devices; (2) Using a 6-DOF motion controller, users can edit tree structures directly with 3D gestures and sketch the 3D tree skeleton. Our interactions are more flexible and convenient than traditional 2D input devices.

2 Related Work

In the past few years, a lot of tree modeling approaches have been proposed to create models efficiently. Trees can be generated automatically using approaches such as rule-based procedural modeling [3, 17], point clouds or image-based methods [2, 10, 21, 24]. Prusinkiewicz et al. presented several L-systems for modeling various tree types with different rules and parameters [15, 16]. However, designing appropriate rules for specific species is quite difficult and complicated. To ease this, Deussen and Lintermann [4, 9] developed the Xfrog system that combines the strength of rule-based approaches with geometric modeling. Runions et al. [19] distribute points randomly in a 3D envelope and use these points to model trees using a space colonization method. Pałubicki et al. [14] extend the space colonization algorithm and present two kinds of light allocation mechanisms for generating more realistic trees. Hu et al. [6] model trees in batches from the airborne LiDAR point clouds. These point clouds are segmented into individual trees using normalized cuts. Subsequently, the tree skeletons are reconstructed from the points using a bottom-up greedy algorithm. Guo et al. [5] apply a procedural modeling technique for modeling trees from multi-view photographs. Wang et al. [22] generate novel 3D tree model variations from existing ones via geometric and structural blending. All these methods, however, do not allow to meet artists' requirements, since they work automatically.

To control tree structures more effectively, interactive tree modeling has been developed. Okabe et al. [12] use a 2D sketch of a tree and generate a 3D tree model by expanding the distances between the branches as much as possible. Tan et al. [20] designed a sketching method to reconstruct the tree model from a single image. Several strokes are drawn to mark up leaf regions and visible branches for guiding the synthesis. Chen et al. [1] convert user's freehand sketches into a 3D tree model by a Markov random field based on parameters obtained from a database of tree models. Since all of these sketch-based methods involve complex parameter analysis, their computational efficiency is low. Longay et al. [11] developed a system for modeling trees interactively using procedural modeling with a multi-touch tablet interface. Yang et al. [25] built a selection-and-recommendation system to recommend tree models according to the 2D input strokes. Users can select 'liked' trees that share some similar shape features with the desired one. Quigley et al. [18] simulated tree animation interactively with the influence of fictitious forces, wind, etc. However, interacting with 3D tree models using traditional 2D input devices, such as a mouse, digital pen or tablet is inconvenient. Furthermore, such 2D input devices cannot support a large range of movements and 3D interaction. Onishi et al. [13] obtain the 3D skeleton of a tree directly based on an interactive L-system and some gestures. This method, however, still edits the tree model by changing the string of the L-System instead of manipulating the tree directly.

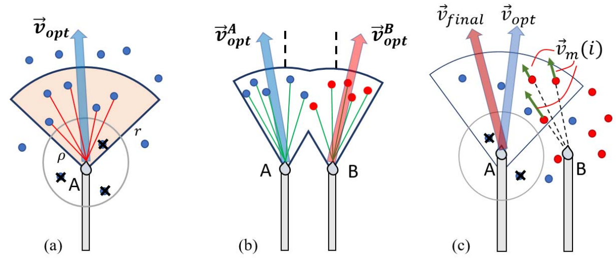

Figure 3: Space colonization. (a) Perception volume and occupancy zone of a bud. (b) Competition for markers between two buds by [14]. (c) Our improved competition strategy.

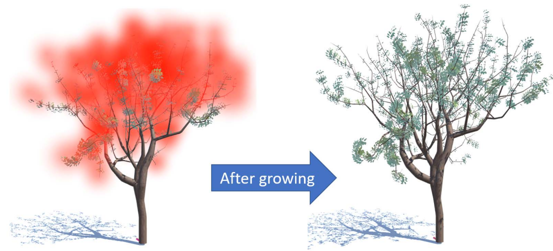

Figure 4: Markers in red are generated only around the alive buds for each iteration.

3 Improved Procedural Tree Modeling

Pałubicki et al. [14] presented a procedural modeling method for tree models based on a space colonization algorithm. In their approach, the space available for growth is represented by a set of marker points. Markers are generated in the whole tree volume using a uniform random distribution. Similar to the concept of multi-way trees as data structures, they describe a tree as a hierarchically bottom-up organized structure [3]. The connection between two nodes is called

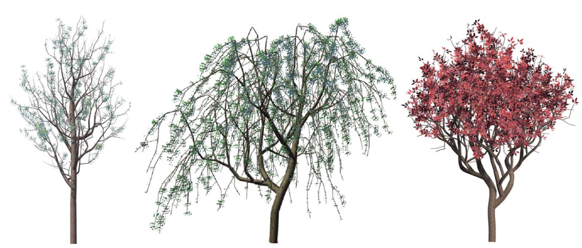

Figure 5: Tree models generated by our procedural method using different parameters.

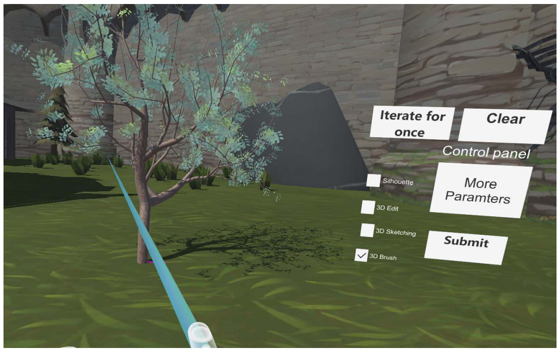

Figure 6: Our system interface with the control panel and pointer.

internode. An internode associated with several leaves and buds forms a *metamer*. Each active bud may generate a sequence of metamers if there is space available. The method assumes that each bud is surrounded by a spherical occupancy zone of radius ρ and has a conical perception volume characterized by the perception angle θ and radius r (see Fig. 3 (a)). We set these default parameters to $\rho = 2$, $r = 4$ and $\theta = 75°$. If a marker is competed by several buds at the same time, it will be associated with the closest bud (see Fig. 3 (b)). The growth direction of a bud can be written as: $\vec{v}_{opt} = \frac{1}{N}\sum_i^N \vec{v}_i$, with $\vec{v}_i$ a normalized vector formed by the bud and the marker position i.

As shown in Fig. 3 (b), buds tend to generate branches far from each other, which is effective to avoid intersections. However, it's costly to associate each marker to the closest bud. In original algorithm [14], for each marker, all close buds within radius r are found at first, so that a large number of distances between markers and buds have to be calculated. The closest one can be determined from such buds near the marker. Moreover, the required distribution of many markers within the tree space results in a huge point cloud and complex computations.

To model trees faster in VR devices, we improve the space colonization method using weighted and guided markers. At first, for each iteration, each active bud will generate a fixed number of markers around it within distance $\kappa \cdot r$ ($\kappa = 2$ *to* 4), as shown in Fig. 4. When a bud generates a new marker i, we calculate a direction $\vec{v}_m(i)$ formed by i and the bud. Moreover, the weight ω_i of marker i can be obtained from the normal distribution, that is, $\omega_i = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_i^2}{2\sigma^2}\right)$, where $\sigma = 1$, and d_i denotes the distance between i and its parent bud. As shown in Fig. 3 (c), the blue and red markers are generated by bud A and B, respectively. Instead of associating markers to the closest bud, the direction $\vec{v}_m(i)$ will also let bud A tend to grow far away from B. After integrating gravity, the final growth direction $\vec{v}_{final}(A)$ of bud A can be written as follows: $\vec{v}_{final}(A) = norm\left(\vec{v}_{opt} + \sum_i^M \omega_i \vec{v}_m(i) + \lambda \vec{v}_g\right)$, where M is the number of markers which are located in the perception volume of bud A but generated by bud B. $\vec{v}_g = (0, -1, 0)^{\mathrm{T}}$ is the direction of gravity. By adjusting the modeling parameters, we can create diverse realistic tree models as shown in Fig. 5.

4 VR-based Interactive Modeling

The interface of our system is shown in Fig. 6. We provide a simple control panel to select the operation mode and use a blue ray to indicate the direction of the controller. As mentioned above, our modeling system provides several operations for users to manipulate the tree structure interactively. The operations support 3D editing, 3D sketching, modeling guided by a brush as well as a silhouette. In this chapter, we will introduce each operation in detail. To show our modeling effect more clearly, we will use less or even no leaves in the following chapters.

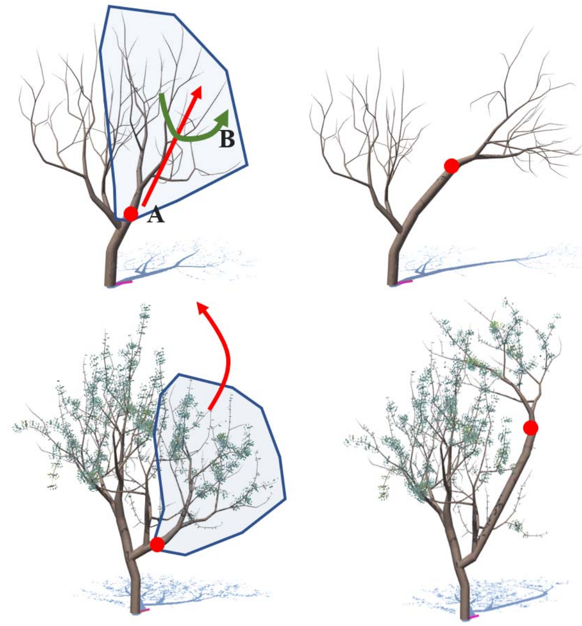

Figure 7: Direct 3D Editing: Simultaneous rotation and translation of a part of a tree using a 6-DOF motion controller.

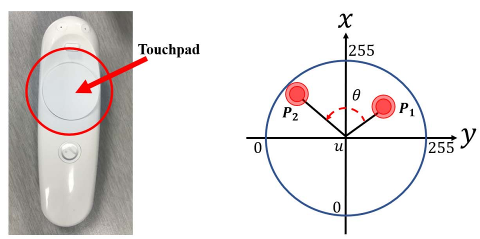

Figure 8: Left: the touchpad of controller; right: precise rotation using the touchpad.

4.1 3D Editing using the controller

With traditional 2D input devices, like mouse or digital pen, the edit operations are often inconvenient and discontinuous for users to design 3D models. For example, user must perform rotation and translation operations separately, and must adjust camera position constantly when manipulating the models in 3D because the operations are limited in 2D. By contrast, the 6-DOF motion controller allows user to manipulate tree models more intuitively and continuously. It's easy to directly edit the model in 3D and simultaneously change both position and orientation of the selected part using the 6-DOF controller (see Fig. 7). The translation and rotation of the controller can be directly mapped to the selected tree part.

Generally, using the transformation of controller directly can roughly meet the requirement for the precision of rotation. However, it's not enough for some high-precision case because the degree of rotating wrist is difficult to control. Thus, we provide an alternative approach for precise rotation using the touchpad of controller. Moreover, this approach is also suitable for some low-level 3-DOF controllers which are still common on the market.

Concretely, user translates the selected part of a tree by moving the controller in space and rotates the part by sliding finger on the touchpad at the same time (see Fig. 8). The angle of rotation $\theta =$

Figure 9: Sketching directly with 3D gesture using the 6-DOF controller.

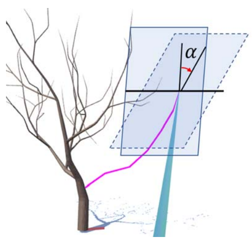

Figure 10: Sketching with depth by rotating the drawing plane using the touchpad.

$\tau \cdot \arccos\left\{\frac{(\mathrm{P}_1-u)\cdot(\mathrm{P}_2-u)}{|\mathrm{P}_1-u||\mathrm{P}_2-u|}\right\}$, where P_1 and P_2 are the positions before and after sliding respectively, $u = (127.5, 127.5)^{\mathrm{T}}$ represents the origin of the coordinate. $\tau = sign\left(\{(\mathrm{P}_1 - u) \times (\mathrm{P}_2 - u)\}.z\right)$ denotes the direction of rotation, here,$\times$ represents the cross product, $\mathrm{P}'_1 = (\mathrm{P}_1.x, \mathrm{P}_1.y, 0)^{\mathrm{T}}$ and $\mathrm{P}'_2 = (\mathrm{P}_2.x, \mathrm{P}_2.y, 0)^{\mathrm{T}}$.

4.2 3D Sketching with the controller

Using the existing sketch-based modeling methods [1, 12, 20], modeler can generate tree skeleton by sketching several 2D strokes. However, 2D strokes cannot reflect the 3D structure clearly. And it's inefficient to infer the 3D skeleton from the 2D intersecting branch strokes.

In this paper, we build a sketch system based on 6-DOF motion controller. With the controller, user can directly draw 3D skeleton strokes using 3D gestures as shown in Fig. 9. In addition, similar to 3D Editing, we also provide an additional version based on touchpad for some inferior 3-DOF controllers. We calculate the distance from the pressed position P to the touchpad center u. Then we mapped the distance into an angle α between 0 to 90 degree, that is: $\alpha = \frac{2\|P-u\|_2}{255} \cdot 90°$. As shown in Fig. 10, angle α is used to rotate a 2D drawing plane around the axis defined by the horizon and current position. User can sketch 3D skeleton by drawing on this plane. As a result, it can achieve a similar effect as 3D sketching using a 6-DOF controller directly.

4.3 3D Brush and Silhouette based modeling

User can guide the growth of a tree by distributing markers using brush or silhouette as in the TreeSketch [11]. However, TreeSketch only provides 2D operations and need multiple fingers for interaction. In this paper, we present a faster and more controllable version to modeling trees interactively using the 6-DOF controller. Our system enables user to brush 3D strokes directly and change the parameters easily with touchpad during moving the controller.

Brushing with 3D gesture and guiding direction. We implement the brush by generating markers along with the strokes drawn by user. Trees will grow within the space defined by these markers (red dots in Fig. 11). However, using procedural modeling approach

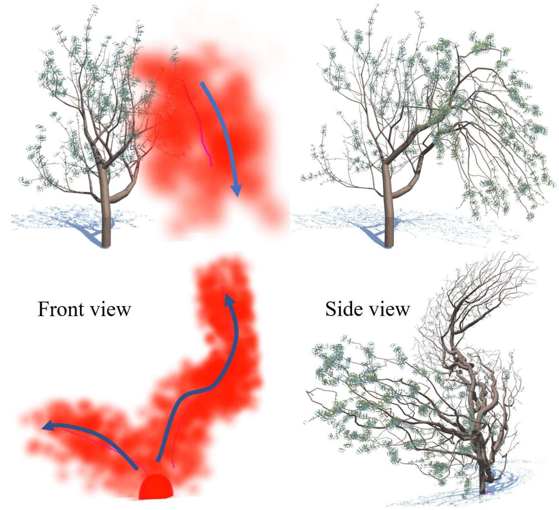

Figure 11: Tree models created by brushing with 3D gesture.

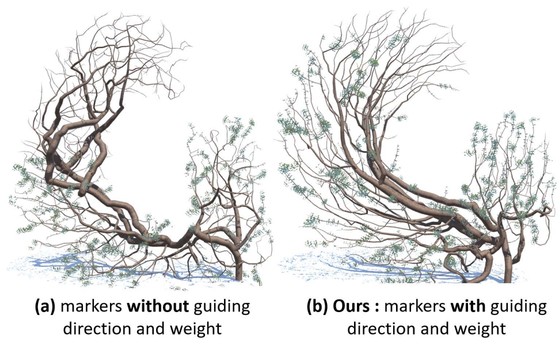

Figure 12: Results without/with the guiding direction of brushes.

only, branches will grow too randomly and intricately as shown in Fig. 12 Left. To make the tree structure more controllable, for each maker i, we first calculate the distance d_i from i to the stroke and record the stroke direction $\vec{v}_{stroke}(i)$ (see Fig. 13). Then marker i will be weighed as ω_i according to d_i using the normal distribution. In this situation, the final bud growth direction $\vec{v}_{final}$ can be described as: $\vec{v}_{final} = norm\left(\vec{v}_{opt} + \sum_i^N \omega_i \vec{v}_{stroke}(i)\right)$. As a result, the branches will grow more fittingly along with the brush strokes as shown in Fig. 12 Right.

In addition, the sketching operation is not limited to a 2D plane. Using the motion controller, a modeler can draw brush strokes with 3D gesture directly. Moreover, some parameters, like the radius of brush and the density of markers, can be changed continuously by the touchpad during brushing the strokes (see Fig. 14).

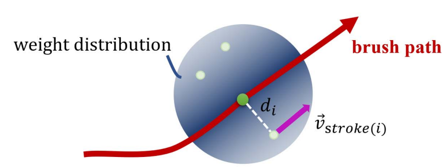

Figure 13: Weight distribution of markers (The darker the blue, the greater the weight).

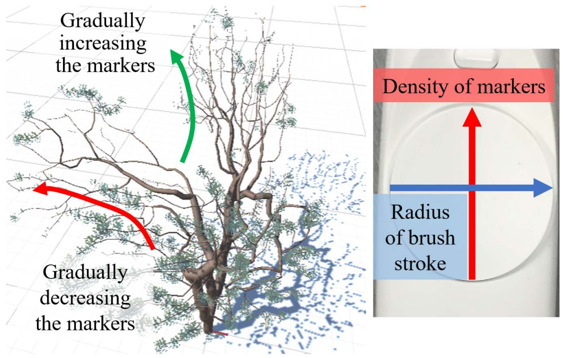

Figure 14: Left: The density of markers can be changed continuously by touchpad during brushing with 3D gesture. Right: How the touchpad controls the two kinds of parameters, including the density of markers and the radius of brush.

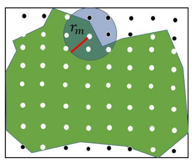

Figure 15: The mask image used for generating markers in silhouette.

Silhouette based modeling. Longay et al. [11] and Wang et al. [23] apply a Teddy-like method [7] to construct a 3D hull from a 2D silhouette sketch, which can be used for guiding the tree growth. However, it will take a lot of time to checking if each marker is inside the hull. In this paper, we describe a simple but faster method to generate markers within silhouette for VR platform. After drawing the silhouette sketch, we first project it into a 2D binary mask image with camera in front. The size of mask image is same as the bounding box of silhouette. Then we traverse pixels one by one and generate marker points randomly only around the masked pixels within a 3D sphere of radius r_m (see Fig. 15). To distribute markers smoothly at the edge, a logistic function is used to describe r_m. Therefore, the r_m at pixel x can be defined as: $r_m(x) = R\left(\frac{1}{1-e^{-kn_x}} - 0.5\right)$, where R is the maximal radius and k determines the degree of how the function is inclined. n_x represents the amount of the unmasked pixel around x within a 5×5 neighborhood. Typical values for these parameters are: $R = 10$ and $k = 0.4$. Then all the markers are transformed back into the original coordinate system easily with an inverse deformation. Finally, the growth of tree can be guided by these markers. Fig. 16 shows some results of our method.

5 Results and Comparisons

In this paper, we developed an interactive modeling system based on Pico Neo integrated machine, which contains an HMD and a 6-DOF motion controller. By alternately using the interactions

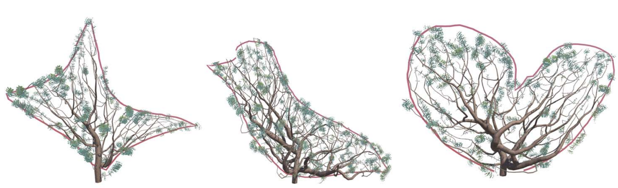

Figure 16: Examples of silhouette-based modeling.

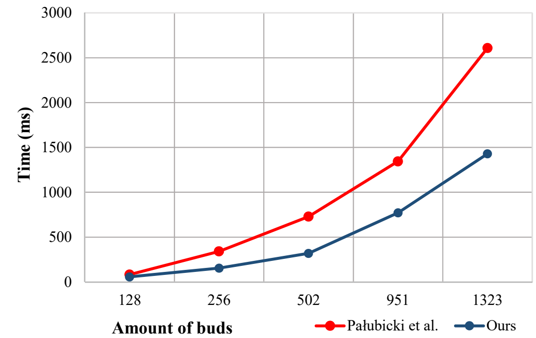

Figure 17: Comparison of computing time for tree modeling using Palubicki's method [14] and our improved space colonization approach. In this experiment, each bud generates about 1300 marker points.

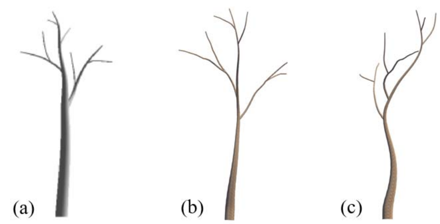

Figure 18: Comparison of trees created by sketches with Chen's method [1]. (a) A tree model created by 2D sketches of [1], the branches of which are generally straight in side view. (b) A similar tree created by 3D gesture directly using our method. (c) Side view of our result. Note that branches can be bent at will through the controller.

provided by our system, it's easy for the modeler to manipulate 3D tree structure and create complex artistic models. For example, as shown in Fig. 19, the modeler first adjusts both position and orientation of a selected tree part, and subsequently brushes two strokes on the model. Compared with existing interactive systems listed in Table 1, our system can directly manipulate 3D structure and provide more flexible and easier operations for user. For example, Fig. 18 shows the comparison with Chen's sketch-based method [1]. Chen's method takes lots of time to infer 3D structure from 2D strokes, and the reconstructed branches are generally straight. Whereas, our method can directly draw 3D branches at will in real time using 6-DOF controller. Moreover, users can change the parameters continuously using touchpad during modeling trees, whereas the parameter adjustment and the modeling operations must be performed separately in most of existing systems.

Additionally, to overcome the shortage of computing resources of VR devices, we present an improved procedural modeling approach. The comparison of computing time is shown in Fig. 17. On the premise of same environment and parameters, our method can achieve same effect as [14] but use less memory and run faster.

6 Conclusion

In this paper, we build an interactive system for modeling trees using a head-mounted display and a 6-DOF motion controller. An improved procedural tree modeling algorithm is first introduced to model trees faster using the guided and weighted markers. In addition, based on a 6-DOF motion controller, the system provides several effective interactions for users to adjust or define the 3D tree structures flexibly. All the interactive operations are implemented after considering the performance of VR devices. Compared with 2D input devices, it's convenient for the user to model trees inter-

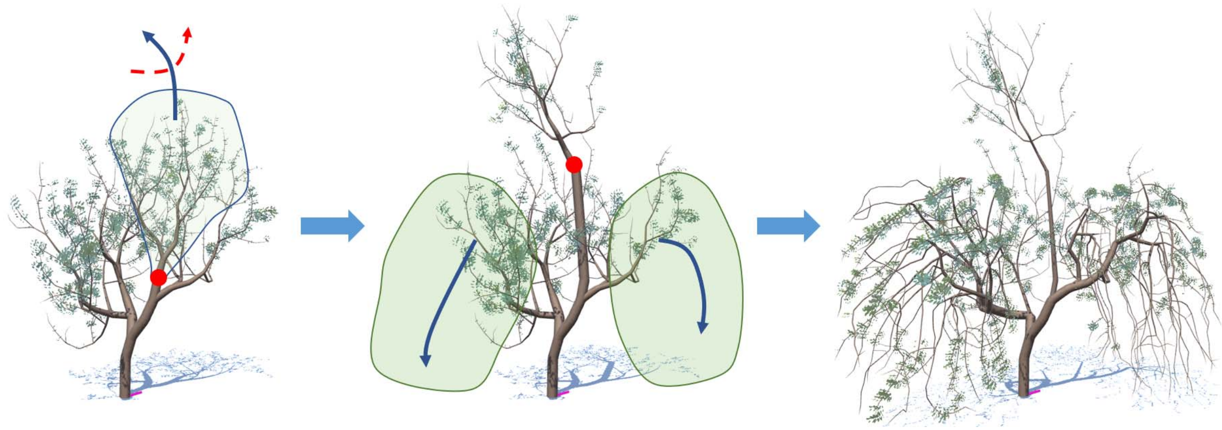

Figure 19: Combination of different interactions. Left: translate and rotate a part of the tree simultaneously. Middle: brush 2 strokes based on current model. Right: final modeling result.

Table 1: Comparisons of the interactive operations between existing tree modeling methods and ours.

Method	Sketch	Edit	Rotation and translation	Brush
Okabe [12]	2D	N/A	N/A	N/A
Tan [20]	2D	N/A	N/A	N/A
Chen [1]	2D	N/A	N/A	N/A
Longay [11]	2D	2D	N/A	2D
Onishi [13]	3D	L-String	N/A	N/A
Our method	**3D**	**3D**	**Simultaneously**	**3D**

actively with 3D gesture input. Using the 3D Edit with motion controller, translation and rotation can be handled simultaneously and continuously, which is hard to be implemented by traditional 2D input devices. Also, our system supports 3D sketching to draw 3D branches directly. Using the 3D-brush and silhouette given by the hand sketch, users can guide the growth directly with a specific shape.

In the future, we would like to design a tree modeling system based on interactions from two controllers rather than one. Additionally, it would be interesting to explore more forms of interactive operations for tree modeling based on motion controller or other VR input devices, like Leap Motion and VR gloves, etc.

Acknowledgments

This work was supported in part by National Key Research and Development Program (2017YFB1002803), and the National Natural Science Foundation of China (61572055, 61633004, 61972388), the CAS Grant (GJHZ1862), the Shenzhen Basic Research Program (JCYJ20180507182222355), the Leading Talents of Guangdong Program (00201509).

References

[1] X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, and S. B. Kang. Sketch-based tree modeling using markov random field. *ACM Trans. Graph.*, 27(5):109:1–109:9, Dec. 2008.

[2] Z. Cheng, X. Zhang, and B. Chen. Simple reconstruction of tree branches from a single range image. *Journal of Computer Science and Technology*, 22(6):846, 2007.

[3] P. de Reffye, C. Edelin, J. Françon, M. Jaeger, and C. Puech. Plant models faithful to botanical structure and development. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, pp. 151–158. ACM, New York, NY, USA, 1988.

[4] O. Deussen and B. Lintermann. *Digital design of nature: computer generated plants and organics*. Springer Science & Business Media, 2006.

[5] J. Guo, S. Xu, D.-M. Yan, Z. Cheng, M. Jaeger, and X. Zhang. Realistic procedural plant modeling from multiple view images. *IEEE transactions on visualization and computer graphics*, 2018.

[6] S. Hu, Z. Li, Z. Zhang, D. He, and M. Wimmer. Efficient tree modeling from airborne lidar point clouds. *Computers & Graphics*, 67:1–13, 2017.

[7] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *Acm siggraph 2007 courses*, p. 21. ACM, 2007.

[8] A. Lindenmayer. Mathematical models for cellular interactions in development i. filaments with one-sided inputs. *Journal of theoretical biology*, 18(3):280–299, 1968.

[9] B. Lintermann and O. Deussen. Interactive modeling of plants. *IEEE Computer Graphics and Applications*, 19(1):56–65, 1999.

[10] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graph.*, 29(6):151:1–151:8, Dec. 2010.

[11] S. Longay, A. Runions, F. Boudon, and P. Prusinkiewicz. Treesketch: Interactive procedural modeling of trees on a tablet. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '12, pp. 107–120. Eurographics Association, Goslar Germany, Germany, 2012.

[12] M. Okabe, S. Owada, and T. Igarash. Interactive design of botanical trees using freehand sketches and example-based editing. In *Computer Graphics Forum*, vol. 24, pp. 487–496. Wiley Online Library, 2005.

[13] K. Onishi, N. Murakami, Y. Kitamura, and F. Kishino. Modeling of trees with interactive l-system and 3d gestures. In *International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, pp. 222–235. Springer, 2006.

[14] W. Palubicki, K. Horel, S. Longay, A. Runions, B. Lane, R. Měch, and P. Prusinkiewicz. Self-organizing tree models for image synthesis. In *ACM Transactions on Graphics (TOG)*, vol. 28, p. 58. ACM, 2009.

[15] P. Prusinkiewicz, M. Hammel, J. Hanan, and R. Mech. L-systems: from the theory to visual models of plants. In *Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*, vol. 3, pp. 1–32. Citeseer, 1996.

[16] P. Prusinkiewicz, M. James, and R. Měch. Synthetic topiary. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 351–358. ACM, 1994.

[17] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, Berlin, Heidelberg, 1996.

[18] E. Quigley, Y. Yu, J. Huang, W. Lin, and R. Fedkiw. Real-time interactive tree animation. *IEEE transactions on visualization and computer graphics*, 24(5):1717–1727, 2018.

[19] A. Runions, B. Lane, and P. Prusinkiewicz. Modeling trees with a space colonization algorithm. In *Proceedings of the Third Eurographics conference on Natural Phenomena*, pp. 63–70. Eurographics Association, 2007.
[20] P. Tan, T. Fang, J. Xiao, P. Zhao, and L. Quan. Single image tree modeling. In *ACM Transactions on Graphics (TOG)*, vol. 27, p. 108. ACM, 2008.
[21] P. Tan, G. Zeng, J. Wang, S. B. Kang, and L. Quan. Image-based tree modeling. *ACM Trans. Graph.*, 26(3), July 2007.
[22] G. Wang, H. Laga, N. Xie, J. Jia, and H. Tabia. The shape space of 3d botanical tree models. *ACM Transactions on Graphics (TOG)*, 37(1):7, 2018.
[23] R. Wang, Y. Yang, H. Zhang, and H. Bao. Variational tree synthesis. In *Computer Graphics Forum*, vol. 33, pp. 82–94. Wiley Online Library, 2014.
[24] H. Xu, N. Gossett, and B. Chen. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. Graph.*, 26(4), Oct. 2007.
[25] Y. Yang, R. Wang, H. Zhang, and H. Bao. Exploretree: Interactive tree modeling in semantic trait space with online intent learning. *Graphical Models*, 91:39–51, 2017.