

Motion Capture and Estimation of Dynamic Properties for Realistic Tree Animation

Shaojun Hu¹, Peng He¹, and Dongjian He²(✉)

¹ College of Information Engineering, Northwest A&F University, Xianyang, China

² College of Mechanical and Electronic Engineering,
Northwest A&F University, Xianyang, China
hdj168@nwsuaf.edu.cn

Abstract. The realistic animation of real-world trees is a challenging task because natural trees have various morphology and internal dynamic properties. In this paper, we present an approach to model and animate a specific tree by capturing the motion of its branches. We chose Kinect V2 to record both the RGB and depth of motion of branches with markers. To obtain the three-dimensional (3D) trajectory of branches, we used the mean-shift algorithm to track the markers from color images generated by projecting a textured point cloud onto the image plane, and then inversely mapped the tracking results in the image to 3D coordinates. Next, we performed a fast Fourier transform on the tracked 3D positions to estimate the dynamic properties (i.e., the natural frequency) of the branches. We constructed static tree models using a space colonization algorithm. Given the dynamic properties and static tree models, we demonstrated that our approach can produce realistic animation of trees in wind fields.

Keywords: Motion capture · Kinect · Dynamic property · Tree

1 Introduction

The realistic modeling and animation of vegetation is a significant problem because of the inherent complexity of plants. The reconstruction of static tree models from images and point clouds has been widely studied; however, there are few studies that explore the scheme for simulating the swaying of trees in a wind field. Recently, motion capture has been extended from tracking human movement to simulating plant motion. A passive optical system has been used in tree and maize motion capture [10, 17], and realistic animation can be achieved using the motion capture data. However, an optical system is expensive and motion capture is limited for indoor trees. In contrast to an optical system, video-based motion capture uses a camera, which has the advantage of low-cost and portability [3, 4, 15]. Although all the aforementioned motion capture and animation methods generated reasonable results, the studies did not explore the vibration relationship of branches.

In our work, we use Kinect to capture static tree point clouds and tree motion in a pulling and releasing experiment, and then study the motion trajectory and dynamic properties (i.e., natural frequency) of branches. Based on a single-view point cloud of a tree captured by Kinect, we use a space colonization (SC) algorithm to reconstruct a static tree model. Then, we calculate the relative rotation angle between a parent branch and a child branch, and extract the dynamic properties of branches using a fast Fourier transform (FFT). Finally, we generate tree animations using the static tree model and the extracted parameters. In this paper, we address the problem of outdoor motion capture by taking advantage of a low-cost depth sensor, Kinect V2. The contributions of this paper are as follows:

- a low-cost method to capture and estimate the dynamic properties of a real-world tree with hierarchical structures; and
- a physics-guided model to animate trees in wind fields using the extracted properties and static tree models.

2 Related Work

Dynamic Property Estimation. Tree dynamic properties play an important role in branch pruning and vibration harvesting. In the early years, the dynamic properties of trees were measured using the relationship between the height and diameter at the breast (DBH) of tree branches [11, 12] or a strain-stress data logger [8]. In the past decade, researchers have used computer vision based methods to measure the dynamic properties of trees. Sun et al. [15] used video clips of tree motion to extract parameters, and used those parameters to synthesize the motion of an artificial tree model. Long et al. [10] used a passive optical motion capture system to capture the motion of an indoor tree in a wind field, and extracted the wind field to drive the motion of the reconstructed model. Both Sun et al. and Long et al. extracted parameters based on force-displacement measurement, whereas in our work, we propose extracting parameters based on the rotation angle. Wang et al. [16] used three synchronized Kinect V1s to capture the motion of indoor potted plants, and used FEM to estimate Young’s modulus and the damping coefficient. Unlike the work of Wang et al., we use Kinect V2 to capture the motion of outdoor trees.

Tree Animation. The study of the dynamic tree model began in the late 1990s. The first work that simulated the stochastic motion of trees and grass was by Shiya and Fournier [14]. Because our work is related to data-driven animation, we only discuss the closest works regarding that approach. Diener et al. [3] used video-captured motion data to drive the artificial tree model. Long et al. [10] used the extracted wind field from a three-dimensional (3D) sequence of reflective markers to drive the motion of a captured tree. Although Long et al. and Diener et al. made use of captured data to simulate motion, they did not establish a relation between the extracted parameters and physics-based animation model. Wang et al. [16] obtained Young’s modulus and damping coefficients, and used

an FEM model to simulate the deformation of a small potted plant; however, this is time-consuming for a complex tree structure. The latest work of data-driven tree animation was proposed by Hu et al. [6]. The researchers used a camera or mobile phone to capture the motion of a tree outdoors. In our work, we take advantage of Kinect V2, which can be used outdoors in weak light and simultaneously capture the depth and RGB of tree motions.

3 Overview

Our motion capture and data-driven animation system consists of three parts:

First, we use Kinect V2 to collect the motion of a tree, which is driven by a pulling and releasing experiment. To obtain the rotation angle in 3D coordinates, we project the textured point cloud of tree motion onto the image plane first, and then perform tracking in the image plane. When we obtain the trajectory in a two-dimensional (2D) image, we map it to 3D coordinates inversely, and calculate the rotation angle based on the 3D trajectory. Based on the 3D rotation angle, we use an FFT on it and obtain the spectrum of motion in the frequency domain. Motion capture and parameter analysis are introduced in Sects. 4 and 5, respectively.

When we obtain the dynamic parameters, we use an SC algorithm to generate the skeleton of a point cloud captured by Kinect outdoors. Once the static tree model is reconstructed and the dynamic parameters are acquired, we establish the relation between the dynamic parameters and physics-based animation model, and then synthesize tree motion. We discuss this in Sect. 6.

In Sect. 7, we present the results and analysis of our study and limitations are also included.

4 Motion Capture

Many approaches have been proposed to capture tree motions in wind or using a pulling and releasing experiment [1, 3, 8, 10]. The main equipment includes a strain-stress data logger [8], an electromagnetic tracking system [1], and a camera [3]. Instead of the aforementioned devices, we use a low-cost depth sensor to perform motion capture. Specifically, we record the motion of a tree using Microsoft Kinect V2, which can capture motion outdoors and provide depth of scene. Figure 1a shows our motion capture system, which consists of a Kinect V2 sensor and a desktop PC, which can be used to collect the motion data of a tree outdoors.

We selected two outdoor Magnolia trees with heights and DBH of approximately 2.6 m and 2.4 cm, respectively, and 3.3 m and 4.94 cm, respectively. To reduce occlusion, we conducted the capture in winter on a leafless tree. Because of self-similarity in branching, it is difficult to search for features in motion data. To perform accurate and efficient tracking for a tree with uncertain features is beyond the scope of our work; thus, for simplicity, we pasted red makers with a width of approximately 3 cm on a selected branch. Because of the limitation

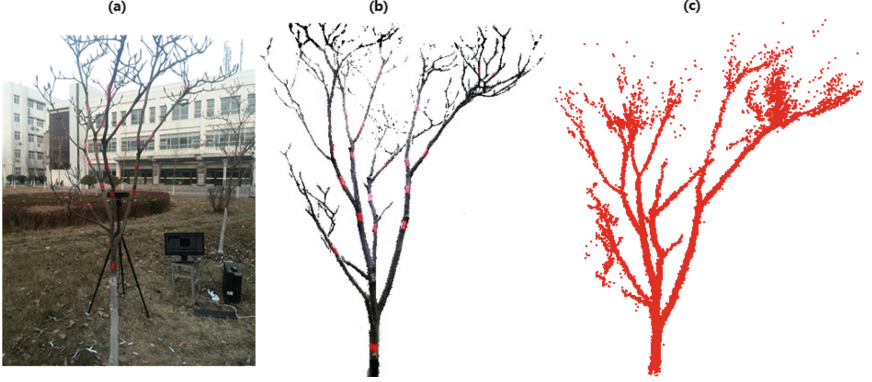


Fig. 1. Scene setting of tree motion capture and captured data using Kinect: (a) capture system; (b) back-projection image of textured point cloud; (c) point cloud.

of the precision of Kinect, it cannot detect tiny branches; thus, we pasted red markers on the first three level of branches, and for levels higher than three, we marked selected parts of them. Tree motion was driven by pulling and releasing branches. To reduce measurement errors, the Kinect direction was set approximately perpendicular to the motion plane.

Unlike video-based motion analysis approaches, our aim is to study tree motion and dynamic parameters in 3D space. However, tracking the motion of branches in a point cloud is difficult; thus, we propose performing 2D tracking first and then mapping the 2D position to 3D coordinates. Hence, not only do we obtain a creditable result, but also reduce the implementation complexity. Figure 1b shows the back-projection of a textured point cloud and Fig. 1c shows the point cloud. From the recorded data shown in Fig. 1c, we observed that the captured data preserved the main branch of the tree in both the back-projection image and original point cloud.

5 Parameter Estimation

5.1 Semi-automatic Tracking

To obtain the motion trajectory and natural frequency of branches, we need to track the motion of branches. Before capturing, we bound red markers on branches to provide a good feature to track. As described in the previous section, we projected the textured point cloud onto a 2D image; thus, we performed 2D tracking first. The mean-shift algorithm is efficient for color-based feature tracking [2] and we tracked a window with a fixed size during the tracking session; hence, we used the algorithm to track the markers on branches. The workflow is as follows:

Step 1: Select the tracked object. Because of the multi-markers in tree branches, we interactively selected a tracking target.

Step 2: Build a tracking model. Based on the CamShift algorithm, we set the tracking trait as a red hue channel, and built a tracking model with a histogram of the selected target.

Step 3: Calculate the probability of the image. According to the histogram of the tracking target, the back-projection of the current frame is calculated.

Step 4: Calculate the mean-shift vector. The mean-shift vector is calculated using the center and centroid of the tracking window.

Step 5: Calculate the stable tracking window. After *Step 4*, we obtain the centroid and mean-shift vector, and move the tracking window to the new centroid along with the mean-shift vector. We repeat *Steps 4* and *5*, and the mean-shift algorithm converges to a stable target area.

Step 6: Track the object continuously. To achieve continuous tracking of the object, we set the next frame, starting with the tracking window as the current frame's stable tracking window, and repeat *Steps 4–6* until all frames are processed.

Constrained by the precision of Kinect and motion blur caused by high-speed movement, the tracking feature becomes weak and tracking may be interrupted. In our work, we adopt two approaches to manage the miss-tracking problem.

The first approach is to consider frames with a weak tracking feature. We set the centroid of the tracking window at the current frame interactively when there were no features to track in the tracking window of the current frame.

To solve the problem of tiny branches with movement beyond the pre-designed search area, we increased the search area and then searched the target iteratively from four directions. Because of the locality of movement and continuous characteristic of the motion sequence, when increasing the search window, the target would definitely be tracked continuously.

Figure 2 shows the tracking results of one branch in selected frames, where the blue box represents the initial window position and the red box represents the stable tracking results.



Fig. 2. Continuous frames of the tracking results of a marker using the mean-shift algorithm.

5.2 Motion Trajectory of a Branch

The motion trajectory of branches is complicated and unpredictable. James et al. [8] captured the motion of a branch using a strain and stress data logger in the north and west directions only, and synthesized the motion trajectory in the plane through this two-direction displacement. In our work, we obtained motion in the plane of branches using mean-shift tracking; however, motion in the plane cannot display the realistic trajectory of a branch. In a motion capture session, we obtained the textured point cloud by combining the image frame and corresponding depth, and projected the textured point cloud onto the image plane. To obtain the 3D position of an image pixel in the projection image as seen in Fig. 3, we retrieved a corresponding image from the point cloud according to the centroid, avoiding re-executing the coordinate transformation of Kinect based on the original depth.

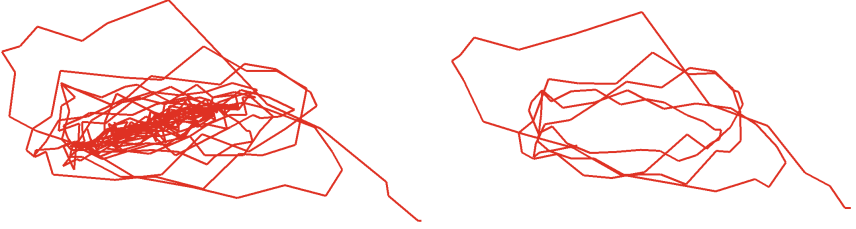


Fig. 3. Trajectory of a branch: (left) front view of the trajectory of the selected branch; (right) the first 80 frames of the selected branch.

5.3 Dynamic Property Estimation

Natural Frequency. Thus far, we have obtained the tracked feature position (3D) along frames. In each frame, we built an approximate hierarchical structure by connecting the feature positions represented by the tracked feature. Mapping the motion of branches to the approximate tree structure and calculating the rotation angle between a parent branch and a child branch rather than using displacement, we obtained the motion of a branch. Considering that the movement of branches is continuous and interrelated, the displacement of the sub-branches contains the displacement of the parent branches. To obtain the pure motion of a branch that eliminates the interference of the parent branches, we propose calculating the relative rotation angle to avoid the displacement of the parent branches in the global coordinate system. Therefore, if the parent branches are set as the reference coordinate system, the motion of the sub-branches relative to the parent branches can be obtained easily.

Figure 4 illustrates our relative rotation angle calculation approach. When pulling and releasing the tree, we obtained the i^{th} frame, as shown in Fig. 4b, where the red arrow indicates the direction of the parent branch and θ_i represents rotation angle at the i^{th} frame between the parent branch and the child branch. The calculation of θ_i uses a plain vector operation and triangular calculation principle described as follows:

$$\theta_i = \arccos \left(\frac{\mathbf{v}_{(AC)_i} \cdot \mathbf{v}_{(CD)_i}}{\|\mathbf{v}_{(AC)_i}\| \|\mathbf{v}_{(CD)_i}\|} \right). \quad (1)$$

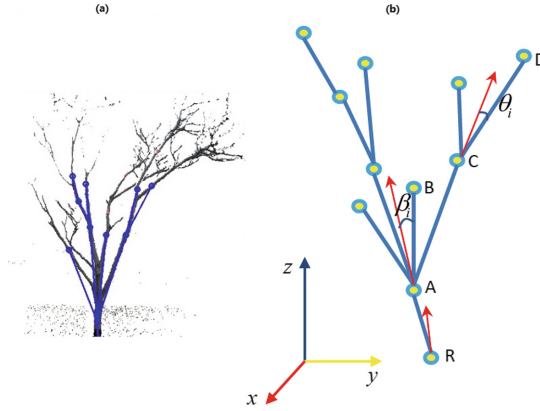


Fig. 4. Illustration of the rotation angle between the child branch and the parent branch: (a) approximate tree structure; (b) the i^{th} frame.

We calculated the rotation angle in 3D coordinates using the method shown in Fig. 4 and obtained the rotation angle sequence in the time domain (as seen in Fig. 6). Motion in the time domain cannot reveal the inherent pattern of branch motion. Therefore, we converted the rotation angle sequence to a frequency domain using an FFT (as shown in Fig. 7). From the signal in the frequency domain, we can clearly analyze the inherent properties of motion. Figure 5 shows the hierarchical structure of our simplified tree, where the numbers 1–6 represent force-bearing points and the letters A–I and a–g represent branches.

Figure 6 shows the change in the rotation angles of four selected branches (C, D, I, F in Fig. 5 tree 1) in the time domain after tracking 409 frames and Fig. 7 shows the corresponding natural frequency of selected branches in Fig. 6. From the frequency domain spectrum results, we observed that each selected branch had one or more dominant frequency.

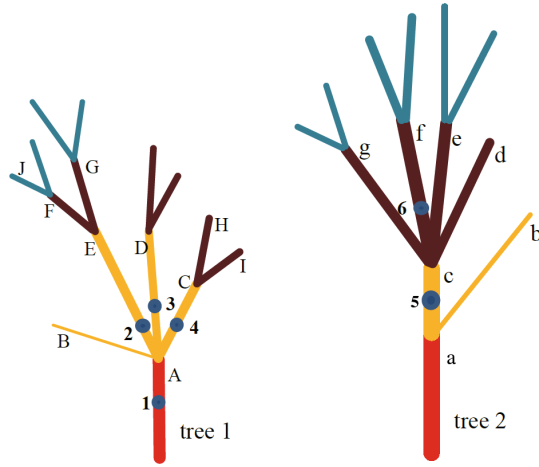


Fig. 5. Illustration of the simplified tree model.

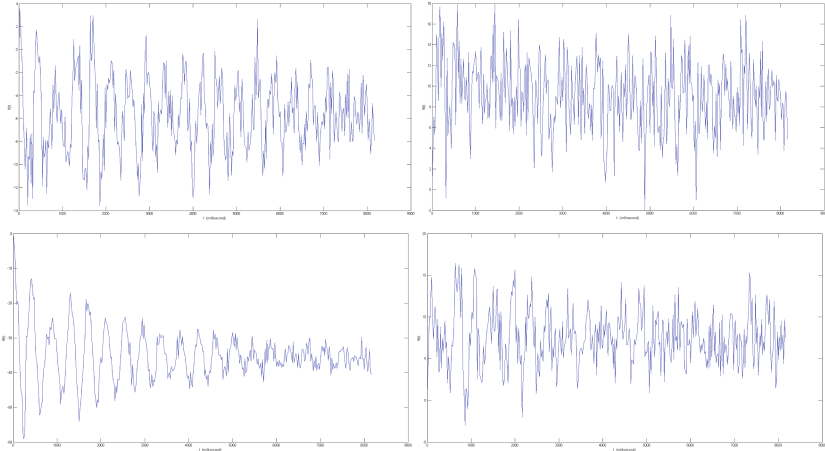


Fig. 6. Changes in the rotation angles: (top) branch C and branch D; (bottom) branch I and branch F.

Damping Ratio. In a pulling and releasing experiment, the movement of the tree stops because of damping. Damping is complicated and consists of several components. To date, the energy dissipation mechanism has not yet been fully elucidated and is usually determined using experimental methods. In practice, researchers relate damping to velocity, and use viscous damping to represent damping in most conditions.

In [7], James proposed several methods to calculate damping: a displacement curve fitting method, logarithmic decay method, and half-band method. Inspired by [15], we derived our damping ratio calculation approach.

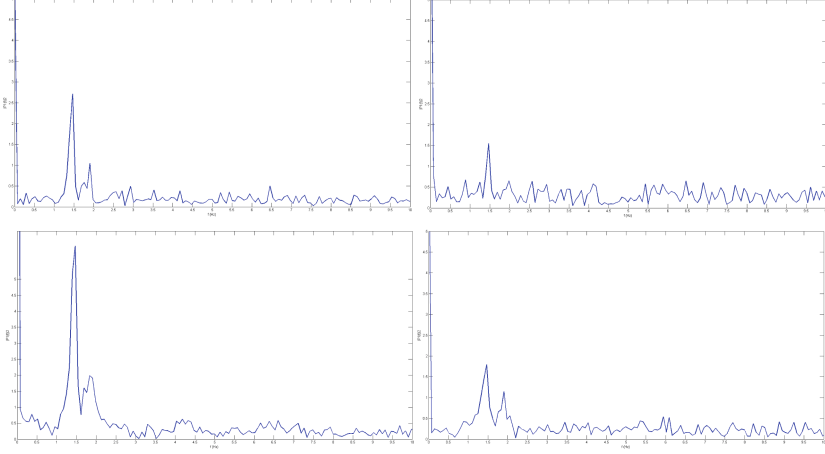


Fig. 7. Natural frequency of branches that correspond to Fig. 6.

In Sun et al.'s [15] paper, damping coefficient γ and angular frequency w satisfy:

$$\begin{aligned}\gamma &= \Delta w \\ \Delta w &= 2\pi \frac{\Delta v}{NT},\end{aligned}\tag{2}$$

where Δv is a sample interval and N, T represents the total number of samples and periods, respectively. In the frequency domain, angular frequency w and frequency f have the following relation:

$$w = 2\pi f.\tag{3}$$

From the work of [15], we learned that damping coefficient γ and damping ratio ξ satisfy

$$\gamma = 4\pi f\xi.\tag{4}$$

Substituting (3) and (4) into (2), we obtain

$$4\pi f\xi = 2\pi\Delta f.\tag{5}$$

We simplify (5), and derive our damping ratio calculation formula in the frequency domain:

$$\xi = \frac{\Delta f}{2f},\tag{6}$$

where Δf represents the frequency variation after the frequency attenuates to half.

5.4 Pattern of the Natural Frequency

To explore the relationship between the natural frequency of different branches, we conducted three comparison experiments on two Magnolia trees. The three

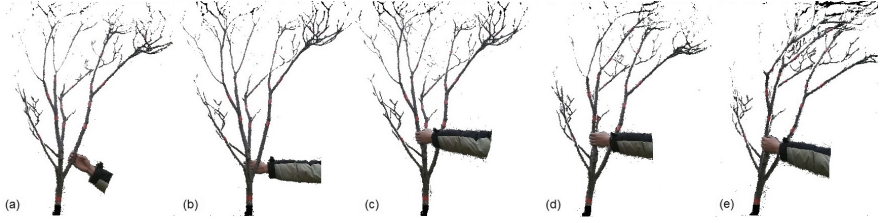


Fig. 8. Different force-bearing point and force applied to branches: (a–c) the different force-bearing points; (d–e) a different force at the same force-bearing point.

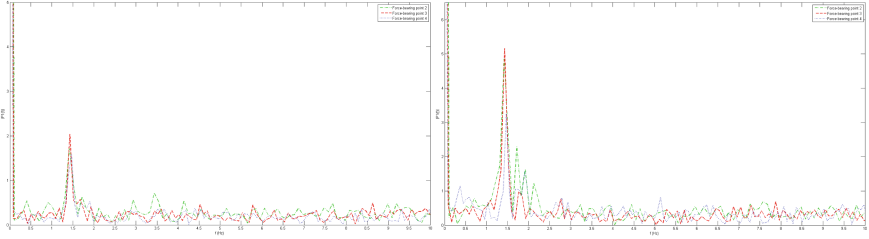


Fig. 9. Natural frequency comparison of the same branch with a force exerted at a different position: (left) branch C; (right) branch I.

experiments were designed as follows: (1) apply approximate force at different positions of different branches; (2) use similar tree species with different tree structures; and (3) apply different forces at the same position of the same branch. Figure 8 shows one frame of motion capture in the comparison experiments.

Figure 9 shows the results for different force-bearing points with an approximate force exerted on them. From the spectrum of the natural frequency, we observed that the first dominant natural frequencies were almost the same (approximately 1.5 Hz). Figure 10 shows the results of different tree structures for one pulling and releasing test. We selected four branches from two trees, and the response spectrum of the vibration shows that the dominant natural frequency of each tree was the same (with tree 1 at 1.5 Hz and tree 2 at 1.2 Hz). Figure 11 shows the results of different forces at the same forced point. Similarly, the natural frequency of different branches was almost the same, but a larger force had a peak value greater than the smaller force.

As shown in Figs. 9, 10 and 11, we can clearly conclude that the first dominant natural frequency of the first three levels of branches were the same.

Figure 12 shows the spectrum of branch J at different forced points. Clearly, high level branches show more complicated vibrations (multiple modal), but they also have in common that they have one dominant frequency near 2.4 Hz.

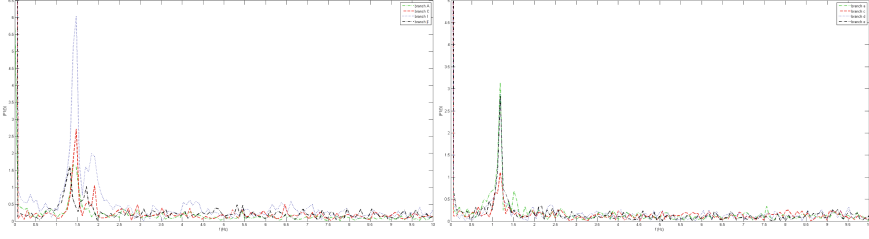


Fig. 10. Natural frequencies of two different tree structures.

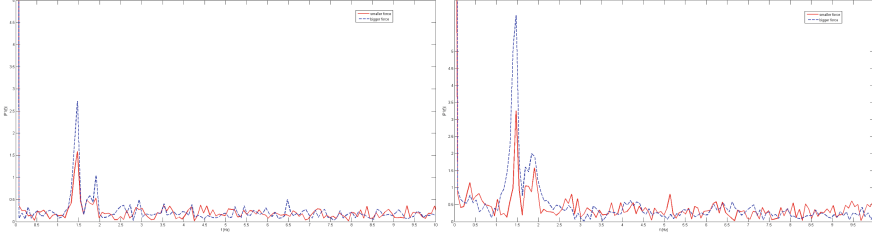


Fig. 11. Influence of different pulling forces on the natural frequency on the same branch: (left) branch A; (right) branch I.

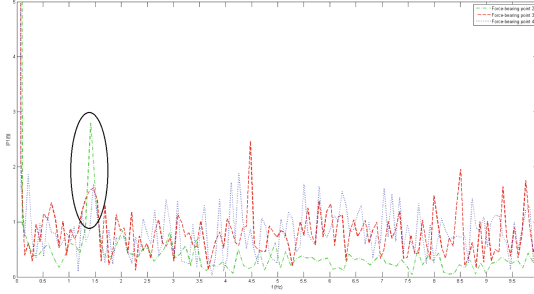


Fig. 12. Spectrum of branch J at different force-bearing points.

6 Tree Modeling and Animation

6.1 Tree Modeling

The generation of tree models is a challenging task which has been studied widely in recent years. In our paper, we chose point cloud based tree modeling because tree point clouds are easy to capture using Kinect. Runions et al. [13] proposed an SC algorithm to generate the tree model of an artificial point cloud and canopy. The SC algorithm resolves branch intersection effectively, and its principle is based on plant growth theory, which is illustrated as competing space for growth between skeleton nodes. In our work, after capturing the tree

point cloud using Kinect, we used an SC algorithm to generate the tree skeleton. Instead of rendering the tree using an L-system, we designed a skeleton node data structure and branch data structure, and constructed a tree hierarchy structured using a self-implemented engine. Finally, we generated a tree geometric model with generalized cylinders, for which the radius of a branch was estimated using the pipe model.

6.2 Tree Animation

Because we obtained the static tree model and dynamic properties of branches, we then used the model and parameters to animate the tree. Many approaches have been proposed to animate tree movement in a wind field. To generate tree motion using a static model and dynamic parameters, we need to determine a feasible physics-based model to implement data-driven tree animation. Hu et al. [4] proposed a tree animation model based on modal analysis, which takes the branch frequency and damping ratio into account. Inspired by Hu et al.'s work, we assumed that a branch was a curved beam and used a simplified physics-based tree animation model.

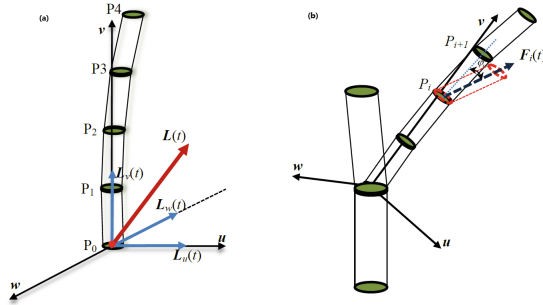


Fig. 13. Our curved branch deformation model.

As seen in Fig. 13a, four segments that consisted of a curved beam (P_0P_1 , P_1P_2 , P_2P_3 , P_3P_4) and located in local coordinate (u, v, w) were subjected to a local net force $F(t)$.

Similar to Hu et al.'s [4] method, we took advantage of modal analysis to establish and solve the dynamic equation of branch motion. From the frequency signal (as seen in Figs. 9, 10 and 11), we assumed that the first dominant frequency had a significant impact on motion. Based on this assumption, we only considered the first dominant mode because the remaining modes were small. The dynamic equation that combines the measured parameters is

$$\ddot{x}(t) + 4\pi\xi f\dot{x}(t) + 4\pi^2f^2x(t) = \frac{|F(t)|}{m}, \quad (7)$$

where f and ξ are the natural frequency and damping ratio of the first mode, respectively, m is the mass branch, and $x(t)$ is the displacement of the branch. We decomposed $F(t)$ into the $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ coordinate, and represented it as $(F_u(t), 0, F_w(t))$ because we considered that the branch did not stretch in the \mathbf{v} direction. Equation (7) can be solved using the explicit Euler method, and we converted the displacement $x(t)$ to an elasticity force according to $\mathbf{F}'(t) = 4\pi^2 f^2 \mathbf{x}(t)$ to control the deformation of the curved branch.

Our final aim was to calculate the rotation angle along with the force direction (as seen in Fig. 13b). Similar to Ref. [5], we converted the curved beam to a spring system and resolved the rotation angle using Hooke's law, which explains that a bending angle is proportional to a bending moment. For further information on the calculation of the rotation angle, see Ref. [5].

7 Results and Limitations

All the modeling and animation tests were performed on a desktop PC with an Intel Core i3 Duo CPU at 3.8GHz and an NVIDIA GeForce GTX 750 video card.

Modeling Results. Figure 14 shows the reconstruction results of the tree point cloud captured by Kinect and reconstructed using an SC algorithm. Comparing the point cloud and generated tree geometric model, we conclude that the tree model preserved the detail of branching and tiny branches, and agreed with the original captured point cloud.

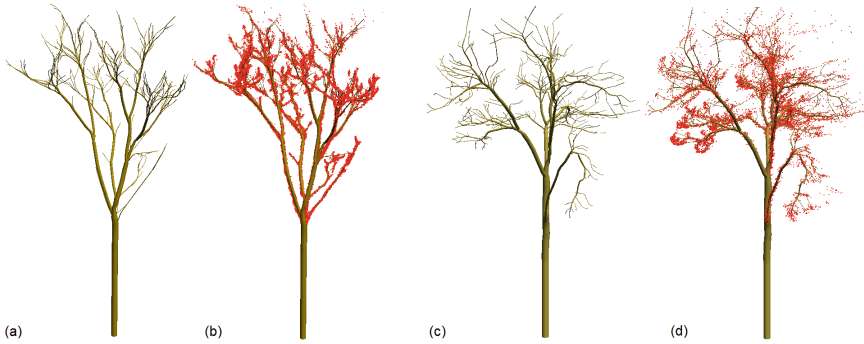


Fig. 14. Reconstructed static tree model using an SC algorithm. (a, c) Two reconstructed tree models; (b, d) the tree models agreed with the point clouds.

Parameter Estimation. Table 1 shows the parameters of the marked branches captured by Kinect. The results show that some branches vibrated at a frequency of 1.39 Hz, but more than 85 % of the branches vibrated at a frequency of 1.46 Hz. From the response spectrum of the vibration, we clearly know that higher level branches demonstrated a more complex vibration modal (as shown in Fig. 12), but it also contained the main stem vibration modal at 1.46 Hz. Based on this

assumption, we only extracted the first mode, which had a natural frequency of 1.46 Hz, and exploited this frequency to build a dynamic equation of branch motion.

Figure 15 shows the damping ratio relation of selected branches corresponding to Table 1. The damping ratios were disorganized; however, we approximately thought that branches with a larger force pulling on them would have a higher damping ratio, with the presupposition that we ignored the measurement error. Because there were fewer captured branches in the reconstructed model, we used the statistical characteristics of the acquired data of some branches to interpolate the damping ratio of the remaining branches. For a more detailed illustration of damping ratio estimation, see Ref. [4].

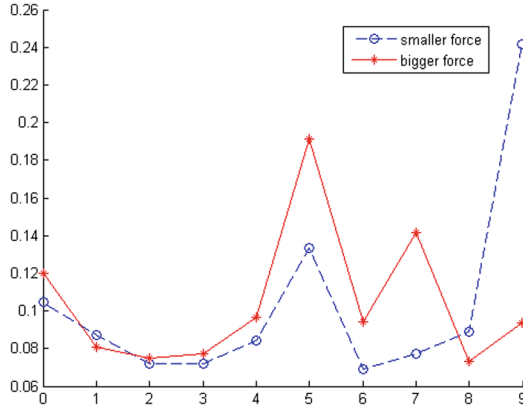


Fig. 15. Damping ratio of branches with different forces.

Animation Results. Figure 16 shows the animation of two Magnolia trees in a wind field generated by our method. The wind field was generated using $1/f^\beta$ noise [9]. One advantage of our physics-based tree model with extracted parameters is the capability to respond to any external force and any model by tuning the parameters.

Limitations. Although we could efficiently reconstruct a static tree model from a Kinect-V2-captured point cloud, constrained by the precision of the device, the detailed branches in the canopy may have been lost. Additionally, during the motion capture session, tiny branches could not be captured because of the precision of Kinect. Second, we only discussed the first three levels of branches' frequency patterns and only used one dominant frequency to generate tree motion. Thus, researching more levels of a branch's natural frequency will be a challenging and interesting issue.

Table 1. Parameters of selected branches.

Branch ID	Small force		Large force	
	f	ξ	f	ξ
A	1.46	0.104	1.46	0.120
C	1.46	0.086	1.46	0.081
D	1.46	0.072	1.46	0.075
E	1.46	0.072	1.46	0.074
B	1.46	0.084	1.39	0.077
I	1.46	0.133	1.46	0.096
H	1.46	0.069	1.46	0.191
F	1.39	0.077	1.46	0.094
G	1.39	0.089	1.46	0.141
J	1.46	0.241	1.46	0.093



Fig. 16. Several frames from the animation of two different trees corresponding to Fig. 14 in a wind field.

8 Conclusion

We proposed a semi-automatic approach to track the markers on a branch and mapped the 2D tracking results to 3D to obtain the 3D trajectory. Based on the 3D trajectory, we derived physical parameters (i.e., the natural frequency and damping ratio) of an outdoor tree. To measure the motion of a branch, we proposed a relative rotation angle principle to calculate the rotation angle of

the branch in local coordinates. To analyze the natural frequency and damping ratio of branches, we converted the vibration of a branch in the time domain to the frequency domain using FFT analysis. Then, we applied the extracted parameters and the static tree model that was reconstructed using an SC algorithm to a physics-based tree animating model. The animation results showed that our approach was feasible for generating realistic tree animations from motion captured data.

Supplementary

An accompanying video can be accessed at Demo link: <http://pan.baidu.com/s/1c1HWZPu>.

Acknowledgment. We thank Dr. Maxine Garcia and Shujie Deng for editing the English text of a draft of this manuscript. The work is supported by the NSFC (61303124) and the Fundamental Research Funds for the Central Universities (Z109021708).

References

1. Adelin, B., Julien, D., Pascal, H., Boris, A., Nicolas, D., Lionel, R., Bruno, M.: A robust videogrametric method for the velocimetry of wind-induced motion in trees. *Agric. For. Meteorol.* **184**, 220–229 (2014)
2. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 564–577 (2003)
3. Diener, J., Reveret, L., Fiume, E.: Hierarchical retargetting of 2d motion fields to the animation of 3d plant models. In: *Proceedings of the 2006 ACM SCA, Switzerland*, pp. 187–195 (2006)
4. Hu, S., Chiba, N., He, D.: Realistic animation of interactive trees. *Vis. Comput.* **28**(6–8), 859–868 (2012)
5. Hu, S., Fujimoto, T., Chiba, N.: Pseudo-dynamics model of a cantilever beam for animating flexible leaves and branches in wind field. *Comput. Animation Virtual Worlds* **20**(2–3), 279–287 (2009)
6. Hu, S., Zhang, Z., Xie, H., Igarashi, T.: Data-driven modeling and animation of outdoor trees through interactive approach. *Vis. Comput.* **33**(6), 1017–1027 (2017)
7. James, K.R.: A dynamic structural analysis of trees subject to wind loading. In: *Phd's dissertation. The University of Melbourne, Victoria, Australia* (2010)
8. James, K.R., Haritos, N., Ades, P.K.: Mechanical stability of trees under dynamic loads. *Am. J. Bot.* **93**(10), 1522–1530 (2006)
9. Khorloo, O., Gunjee, Z., Sosorbaram, B., Chiba, N.: Wind field synthesis for animating wind-induced vibration. *Int. J. Virtual Reality* **10**(1), 53–60 (2011)
10. Long, J., Porter, B., Jones, M.: Animation of trees in wind using sparse motion capture data. *Vis. Comput.* **31**(3), 325–339 (2015)
11. Moore, J.R., Maguire, D.A.: Natural sway frequencies and damping ratios of trees: concepts, review and synthesis of previous studies. *Trees* **18**(2), 195–203 (2004)
12. Moore, J.R., Maguire, D.A.: Natural sway frequencies and damping ratios of trees: influence of crown structure. *Trees* **19**(4), 363–373 (2005)

13. Runions, A., Lane, B., Prusinkiewicz, P.: Modeling trees with a space colonization algorithm. In: Proceedings of the Third Eurographics Conference on Natural Phenomena, NPH 2007, pp. 63–70. Eurographics Association, Aire-la-Ville, Switzerland (2007)
14. Shinya, M., Fournier, A.: Stochastic motion-motion under the influence of wind. *Comput. Graph. Forum* **11**(3), 119–128 (1992)
15. Sun, M., Jepson, D.A., Fiume, E.: Video input driven animation (vida). In: 9th IEEE International Conference on Computer Vision, Nice, France, pp. 96–103. IEEE (2003)
16. Wang, B., Wu, L., Yin, K., Ascher, U., Liu, L., Huang, H.: Deformation capture and modeling of soft objects. *ACM Trans. Graph.* **34**(4), 1–12 (2015)
17. Xiao, B., Guo, X., Zhao, C.: An approach of mocap data-driven animation for virtual plant. *IETE J. Res.* **59**(3), 258–263 (2013)

Next Generation Computer Animation Techniques
Third International Workshop, AniNex 2017,
Bournemouth, UK, June 22-23, 2017, Revised Selected
Papers

Chang, J.; Zhang, J.; Thalmann, N.M.; Hu, S.-M.; Tong, R.;
Wang, W. (Eds.)

2017, VIII, 249 p. 148 illus., Softcover

ISBN: 978-3-319-69486-3