

Research Article

Reconstructing 3D Tree Models Using Motion Capture and Particle Flow

Jie Long and Michael D. Jones

Department of Computer Science, Brigham Young University, Provo, UT 84604, USA

Correspondence should be addressed to Jie Long; may109@gmail.com

Received 8 May 2013; Revised 27 August 2013; Accepted 18 September 2013

Academic Editor: Alexander Pasko

Copyright © 2013 J. Long and M. D. Jones. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recovering tree shape from motion capture data is a first step toward efficient and accurate animation of trees in wind using motion capture data. Existing algorithms for generating models of tree branching structures for image synthesis in computer graphics are not adapted to the unique data set provided by motion capture. We present a method for tree shape reconstruction using particle flow on input data obtained from a passive optical motion capture system. Initial branch tip positions are estimated from averaged and smoothed motion capture data. Branch tips, as particles, are also generated within a bounding space defined by a stack of bounding boxes or a convex hull. The particle flow, starting at branch tips within the bounding volume under forces, creates tree branches. The forces are composed of gravity, internal force, and external force. The resulting shapes are realistic and similar to the original tree crown shape. Several tunable parameters provide control over branch shape and arrangement.

1. Introduction

Reconstruction of tree shape from motion capture data is an important step in replaying motion capture of trees under external forces, such as natural wind. Motion capture provides a fast and easy way to collect the locations over time of retroreflective marker locations placed on an object. In this paper, we address the problem of creating 3D tree shape from motion capture data. We also discuss the best practices for collecting motion data from a tree. This research will focus on reconstructing static 3D tree shape with branching skeletons from data collected by motion capture system.

Solutions to the motion capture problem for trees can be applied to problems in visual effects and the study of tree motion. Motion capture is a potential solution because motion capture data includes wind effects, which are difficult to model in simulation, such as variable branch stiffness, nonuniform variation in size, and emergent effects due to leaf deformation.

Tree shape modeling has long been studied on computer graphics. L-systems [1–3] generate branching structures using predefined rules. Parametric models [4] embed tree's biology information into growth and shape using a parametric set.

Approaches based on photographs [5–7] or videos [8, 9] create tree shapes in 3D space by image processing methods. Laser scanning has been employed for collecting information of 3D tree shapes [10–12]. Particle systems [6, 13–15] represent each branch as a result from particle flow simulation.

Most of these methods result in satisfying tree shapes but do not leverage the 3D positions of motion capture markers, which are recorded as part of a motion capture session but require a different set of inputs. Image- or video-based approaches convert a set of 2D input images into 3D tree models by filling in the missing dimension. Motion capture systems can record tree shape in 3D with high precision (using similar techniques for converting a set of 2D images to a 3D model). Prior work in reconstructing tree shape from motion capture data uses either exact measurements or markers placed within the crown. The approach based on physical measurements can reconstruct the tree shape for a single measured tree, but our method scales to produce multiple tree shapes from the measured data. In the previous application of recording tree shape from motion capture, markers are placed within the crown and the visibility of these markers could be occluded easily with tree leaves. We solve the problem by designing a new process for motion capture

in which we place markers on the edge of the tree crown to maximize the visibility of markers.

We reconstruct 3D tree shape using particle flow with motion capture data as input. Passive optical motion capture (<http://www.naturalpoint.com/>) records locations of reflective markers in the capture arena. We place markers only at branch tips on the edge of the tree crown. Approximately 30 markers cover the crown shape of a medium sized tree. We do not put markers on each branch tip because passive optical motion capture systems cannot reliably track more than 70 markers due to device limitation. A particle flow algorithm generates branching structures starting at the recorded tip positions. Additional starting points are defined within the estimated volume of the tree crown using a vertical stack of bounding boxes or a convex hull. The bounding space approximates the tree crown and bounds particle flow and creation. The step length of a particle's flow varies with the distance to the nearest trunk point. The direction of particle motion is a combination of three forces, which are gravity, internal force, and external force. In our research, we define a force called shape-format as an instance of internal force and apply wind force as external force. The shape-format guides the particles to preserve the original tree shape. The dominant wind direction is recorded during the motion capture process. We also introduce two vectors with one pointing to the nearest trunk point and the other pointing to a constant predefined attractor point. These vectors are factors for the direction and magnitude of the forces.

Compared with the existing work, we present a new procedure of extracting 3D tree shape with branching structures from motion capture system and prove the following advantages over the existing work. First, the functions guiding particle flow are classified into three categories, which are extensible for adding more features and are not limited to functions that the paper is presenting, to produce various tree shapes as desired. Theoretic knowledge, including biology and dynamic mechanism about trees, is embedded into the functions. Second, our approach does not require preknowledge of rules, such as the rule-based approach or parametric models. Third, motion capture provides more information about the original natural tree shapes than the existing data collection process. Photographs provide 2D data for the original tree shape while other approaches, such as the rule-based approach or parametric models, might not start from natural tree shapes. With more information available, we produce better visually plausible tree shapes to the original trees. Fourth, our method improves accuracy of the positional information of tree branches as well as improves efficiency of the procedure to generate 3D tree shapes. Photographs- or video-based approaches usually require techniques related to image processing or camera calibration, which are difficult to make accurate. By applying motion capture system, we directly use the industrial standard of these techniques. In addition, our approach extends the research objects of motion capture from popular rigid bodies to nonrigid bodies. Existing research applying motion capture to trees assumes that trees are rigid bodies or semirigid bodies. In our research, we treat trees as complete nonrigid bodies, which better fits trees' nature.

In this paper, we propose a new particle flow method for reconstructing tree shape from only motion capture data. Our primary contributions are

- (i) a simplified particle flow algorithm for constructing tree shapes from a sparse set of branch tip positions as collected as part of motion capture,
- (ii) a method for deploying passive optical motion capture to reconstruct the shape and motion of trees in the laboratory.

This paper extends our prior work on 3D tree modeling from motion capture [16]. In this extended paper, we generalize the forces that guide particle flow. The generalized forces can be customized in the framework of forces to produce different kinds of 3D tree models from the same motion capture data. In addition, we implement the convex hull as an alternative approach to bounding all the particles and marker locations on a tree. This paper includes 3D models of trees built using these extensions and the models are compared to prior work.

The combination of motion capture with a particle flow method provides an innovative way of creating 3D tree shapes. The resulting tree shapes are similar to the original trees, as shown in Figure 1, and can be used to replay motion similar to the captured motion. With the flexibilities of tuning the forces, we can produce several tree shapes besides the original shape. Figure 2 shows various tree shapes generated from the same set of motion capture data and demonstrates the flexibility of our forces for guiding particle flow.

2. Related Work

Our work is most closely related to prior work in tree modeling and applied motion capture. Our purpose is to investigate methods for creating tree shapes on which motion capture data can be replayed. Compared to prior work in modeling trees, we introduce new equipment for collecting shape information of trees and improve the particle flow process by creating a new set of functions to compute attracting factors. Specially, the particle flow process fits well with the data collection process. Compared to prior work in motion capture, we design a new data collection process for nonrigid bodies and reconstruct realistic 3D models out of the data.

2.1. Tree Modeling. L-systems [1] generate a tree's branching structure using axioms and rules in a concurrent context-free rewriting system. L-systems have been extended in many ways. The extension most relevant to our work is [3] in which L-systems are enriched with partial differential equations and can be parameterized to reconstruct the shape of a specific tree or plant. However, L-systems require understanding of the rewriting system and the according parametric systems. With different rules and parameters, the resulting shapes of trees could greatly vary. Most of the time, L-systems build a tree model from trunk to branches. For our work, we collect positions of branch tips for describing tree shape. Starting from these positions of branch tips back to the trunk, it

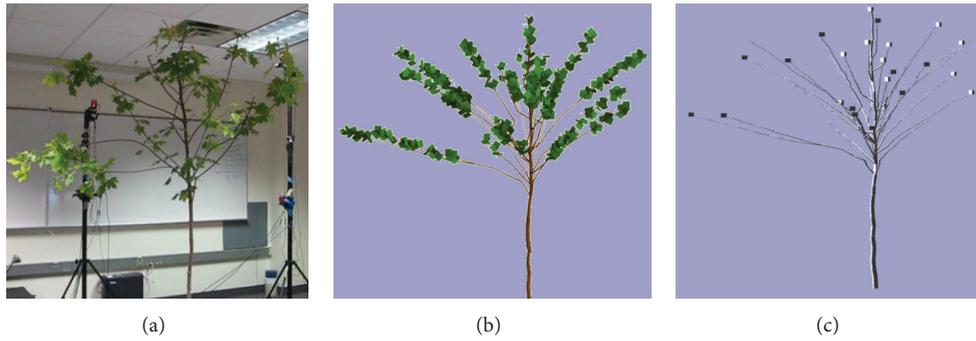


FIGURE 1: Maple model. (a) Is a photo of the original tree. (b) Is the 3D tree model from the result. (c) Demonstrates the branching structure and locations (in grey cubes) where markers are placed.

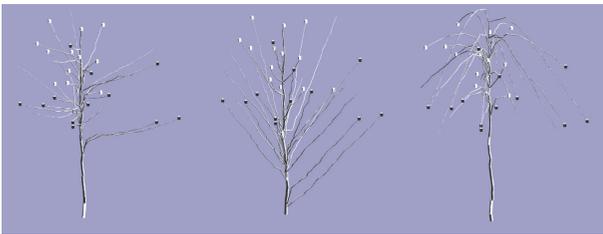


FIGURE 2: Various tree shapes generated from the same set of motion capture data.

becomes harder to combine the collected data from motion capture with L-system techniques for generating 3D tree shapes.

Laser scanning can be used to record 3D tree shapes. Prior work such as [10–12] collects tree shape information from laser scanning and reconstructs 3D tree shapes. Livny et al. [11] start from the position of a tree's root and recursively refine a BSG graph (Branch-Structure Graph) given a set of global parameters. Later Livny et al. [12] further develop an approach which creates a lobe-based representation of the tree crown and adds texture using a predefined branch library for different shapes and species. The texture based tree models improve the visual quality of 3D tree models but make it difficult to create tree animations because it is hard to animate trees with texture clusters. Xu et al. [10] start from the root and use a shortest paths based approach to create branching structures. Laser scanning records about 50,000 points per second and thus collects more complete information about tree shapes. It also has the advantage of recording bigger trees. However, laser scanning records leaves and branches together in the same point cloud. That mixes positional information and requires extra effort to separate leaves and branches. Leaf occlusion makes it hard to record branching structures when leaves have high density, especially in the crown area or in certain tree species. With more positional information, the point cloud gets big and requires more computational resources during processing. It is hard to keep track of a single point in the point cloud among frames and this limits its application in recording tree animation. In our research, we place markers on the edge of the tree crown, which preserves the tree crown shape.

As laser scanning could not address the exact branching structure even with more available information about the tree's shape, our research is not aimed at getting the exact branching structure, but, instead, we provide flexibility to users who adjust the parameter values for generating various branching structures from the same set of recorded data. Mocap records less information about tree shape, but the data is very accurate. For moving trees, we could easily match a point position in its neighboring recorded frame because the recording rate is about 100 frames per second. The design allows researchers to further explore the usage of mocap in animating trees and this is where we contribute a method to model 3D trees from mocap as the first step of tree animation.

More recent work in tree shape modeling involves particle systems. With the exception of Palubicki et al.'s work [15], particle systems approximate tree shapes obtained from photographs [5–7]. Palubicki et al. devised a particle flow which approximates bud fate models. This research extends the Borchert-Honda (BH) model and combines with the concept of self-organizing trees. The simulated amount of light guides the particle flow process and results in different tree shapes. Comparing to Palubicki's work, in our research, we designed a new particle flow process which fits with the positional data collected from motion capture to build a tree model. The particle flow formats the tree's branching structure. Instead of the light amount computed in Palubicki's work, we drive the flow using three forces: gravity, internal force, and external force. In addition, the factors for internal force and external force can be defined differently in different circumstances. Thus, our model has better extensibilities to create trees with various shapes.

These methods are not directly applicable to motion capture data because these methods use photographs or environmental conditions that are not collected during motion capture. Photographs of the tree could be taken during motion capture (and indeed are taken during optical motion capture), but we focus on taking only marker positions as input because this simplifies data collection and processing by reusing the background removal and image alignment performed as part of marker position calculation.

2.2. Applied Motion Capture. Motion capture has been mainly used in rigid bodies, such as human motion. It

produces positions for points on an object over time with very small measurement errors. Our main contribution, regarding motion capture, is that we extend the usage of motion capture from the popular rigid bodies to nonrigid bodies. In addition, we design an innovative data collection process for describing tree shapes using motion capture.

Motion capture systems have been widely used for human or animal motion capture [17–19]. Kirk et al. [20] automatically generate rigid skeletons from optical motion capture systems by preserving a constant distance for each rigid part. These algorithms assume that the distance between markers on the same bone is invariant and cannot be directly applied to nonrigid bodies, such as natural trees, because the distance between markers is not invariant as the object deforms.

Prior work in motion capture for nonrigid bodies includes several approaches to facial motion (including [21, 22]). These methods are based on domain-specific features of the facial structure or patterns. Obviously these domain-specific features do not apply to tree shape reconstruction or motion capture.

The uniform branching structure of human bodies has led to well-understood processes for deploying markers on a human. The number and placement of markers is a critical part of successful motion capture using a passive optical system. Trees have a more complex and less predictable topology and this will require a different approach to marker placement. Previously [23], researchers attempted to directly replay the motion of a tree in wind following the exact motion paths collected for all branches by putting markers on every branch of the tree. Leaves on the tree create marker occlusion which results in poor data. In addition, manually defining branch topology to exactly match the subject tree is labor intensive.

In this paper, we design a data collection and tree modeling process to overcome these difficulties by building a similar, but not exact, copy of the branching structure from a partial collection of branch tip positions. Ongoing work focuses on replaying collected motion such that the motion looks natural on an approximate copy of the branching structure.

3. Motion Capture of Trees

In this section, we describe how to collect data from trees using a motion capture system such that the data supports reconstruction of tree shape. The data will be collected indoors on trees with height less than 2.5 meters. The data collection process results in an unindexed set of marker locations over time for a small set of instrumented tree branch tips.

We use a passive optical motion capture system (OptiTrack V100 by NaturalPoint) to collect data. The passive optical capture system strikes balances conflicting features. Passive optical systems can reliably track up to 70 markers, and some markers weigh only a few grams. This is ideal for working with tree crowns. Active optical and active magnetic systems use heavier markers and cannot track more than 20 markers at once. The magnetic systems have the advantage



FIGURE 3: Marker placement on the crown periphery avoids occlusion while generating data that can be used to generate crown structure.

that they do not have visibility occlusions and can track position and rotation but are more expensive than passive optical systems and can track fewer branch tips. Magnetic system markers are heavier and may alter branch tip motion.

Collecting data from a natural tree is challenging for passive optical motion capture systems because trees are not rigid bodies and partially self-occluding. The following method for deploying a passive optical motion capture system collects data from which tree shape and motion can be inferred.

Twelve or more cameras are arranged in a circle around the tree with 6 cameras located approximately 0.8 meters above the ground and another 6 cameras located 3.3 meters above the ground. For each camera, field of view is adjusted to include the entire tree. About 30 markers are placed on branch tips throughout the crown. Markers are square retroreflective markers with surface area of about 1 cm and adhesive backing. Markers are placed to cover branch tips on each major branch from the stem and to provide nearly uniform coverage of the crown. On these branch tips, the square shaped markers are wrapped to cover the whole tip so these markers are visible to most of the cameras from different view angles. Uniform coverage improves both shape reconstruction and motion capturing. Placing markers such that their motion paths overlap complicates algorithms for extracting motion paths from unindexed marker positions. For a medium sized tree, the number of branch tips exceeds the number of markers so that not every branch tip is covered by a marker. Leaves around the marker are removed to improve marker visibility. So in the resulting 3D tree shape, the shape of the original tree crown will be preserved. While recording tree motion we use an electric fan to create wind around the tree because data is collected indoors. The wind direction is inferred from the location where it sits relative to the tree. Other statistic methods, such as PCA (principle component analysis), can also compute the dominant wind direction after motion data of branch tips are processed.

Photographs in Figure 3 show the arrangement of the motion capture cameras and the reflective markers as deployed on an indoor pine tree. Markers are placed at branch tips shown as white dots in the image. The right side image shows marker location in the red box on the left side image.

Our design requires less user intervention, produces cleaner motion capture data, and may support animation of tree motion.

4. Data Processing

Simply inferring positions from one frame of captured data is not adequate due to noise. These branch tips with markers are called “recorded” or “captured” tips. However, the initial locations of recorded branch tips may contain errors due to noise in either the system or the capture environment.

A clustering algorithm approximates a single initial position from a collection of captured initial positions for recorded branch tips while minimizing error from the motion capture system. The clustering algorithm analyzes positions over many frames of motion captured data and eliminates gaps and noise. Gaps occur when a marker is not present in a frame. This algorithm uses forward differencing to predict a position for a marker at frame M based on positions in previous frames. The closest marker in the next frame is added to the motion trace for the marker if there is a marker located close enough to the predicted position. If there is no marker position recorded close enough to the predicted position, that marker is marked with no data, that is, a gap, for that frame. Gaps are repaired using interpolation over the motion path for a marker. If the number of marker positions recorded for a marker over time is less than 1/4 of the total frames, that partial marker trace is marked as noise and eliminated.

If the capture process includes several hundred frames of data in which the tree is not moving, the averaging branch tip positions over these frames result in precise estimates of initial positions. In most cases, the number of markers inferred from the clustering process matches the number of markers placed on the tree.

5. Generating 3D Tree Shape

Particle flow is a well-studied approach to generating 3D branching structure for trees [6, 13–15]. We adapt the method to motion capture data. The 3D tree crown boundary inferred from motion capture data constrains the particle range and preserves the original 3D tree silhouette. We use a stack of bounding boxes or a convex hull to represent the crown boundary. Particles, either generated randomly in the boundary or locations of branch tips recorded by motion capture, are moving towards trunk nodes. Three forces, from gravity, internal force, and external force, drive the particle flow process. The paths of the particle flow produce branching structure. By attaching leaves to the branching structure, we generate a 3D tree model that has similar appearance as the natural tree shape.

We synthesize a trunk in the center of the crown shape, as shown in Figure 4. Figure 4(a) shows a photograph of pine tree with markers placed on the crown. Figure 4(b) contains a bounding box of these marker locations and a straight vertical line representing trunk shape. The length of the line is scaled by the crown height of the bounding box. On this line segment, we generate about 10 trunk nodes. Random offsets to these nodes in x and z directions are added as shown in Figure 4(c).

A particle represents a branch tip. One group of particles is 3D positions of branch tips recorded from motion capture, which is described in Section 4. Figure 4(a) shows a photograph of a pine tree with markers placed on the crown. All the branch tip locations recorded by motion capture are labeled in white dots. These particles are shown in black circles in Figures 5(c) and 6.

Because of limitations with the motion capture device’s capability, we cannot record location information for every branch tip on the tree. Another group of particles is randomly generated within the bounded space of all the recorded branch tip positions. In Figures 5(c) and 6, these particles are in green color.

Instead of using one single bounding box for the whole tree crown, as shown in Figure 5(a), we create a vertical stack of bounding boxes as shown in Figure 5(b), where the crown height is evenly divided. The number of bounding boxes varies to match the original tree shape and required level of detail. In the case of the pine tree shown in Figure 5, we use four parts which can closely match the tree shape well. Particles are randomly generated inside the smaller bounding boxes, as shown using green dots in Figure 5(c). The total number of branch tips, including motion capture branch tips and randomly generated particles, is set to be similar to the original tree. The number of green dots in each box is proportional to the number of black dots. Therefore, we maintain a similar total amount and distribution of natural tree branch tips.

Alternatively, we use a convex hull for all the particles, as shown in Figure 6. A convex hull provides more precise bounding space than the stack of bounding boxes. However, it requires a (slightly) more complex boundary detection scheme and more implementation details. To pursue better fit for the tree shape, other bounding methods, such as alpha shapes [24], could also be considered.

For the pine tree’s trunk, we generate 9 nodes in a straight vertical line and add random offsets in x and z directions to these trunk nodes. The length of the line is scaled about 1.2 times of the pine tree’s crown height.

Trunk nodes contain a root attractor point and a nearest trunk point, as shown in Figure 7. The root attractor point, shown in green, is the trunk node closest to the lower bound of the bounding box for the entire crown. The nearest trunk point, which is shown in blue for the red particle in Figure 7(a), is the closest trunk node to a single particle.

Particles move under directions of three forces: gravity, shape-format, and dominant wind. In Table 1, we describe direction and magnitude of each force. Gravity points vertically down to the ground. We assume that a particle has bigger mass when it is closer to the trunk. This assumption follows the observation that when closer to the trunk, a branch has a bigger radius. For a particle with bigger mass, it has bigger magnitude of gravity and moves faster in the vertical downwards direction. Under this assumption, we set the magnitude of gravity proportional to the distance between a particle position S_p and the nearest trunk point S_t .

We call the second force shape-format. Arborists distinguish styles of growth habit of trees using different classifications, such as excurrent and decurrent. The force

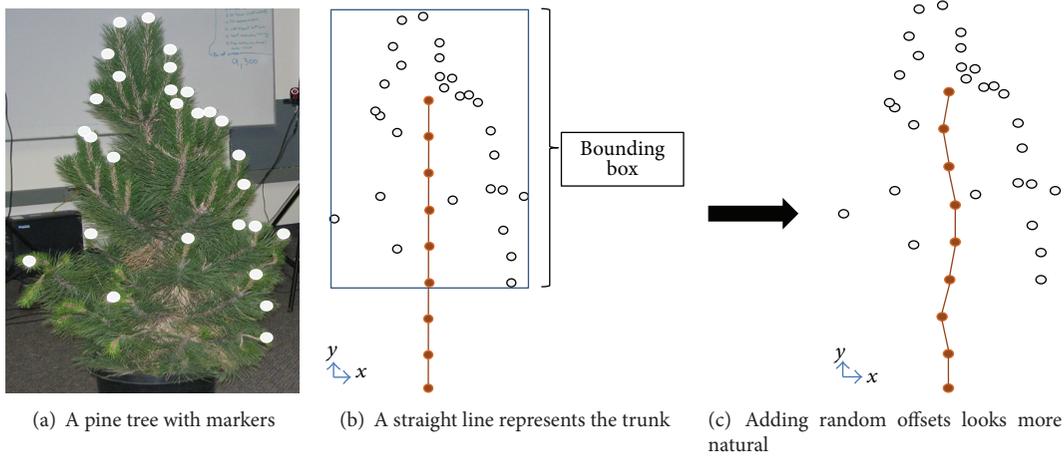


FIGURE 4: A simple trunk model is added to the collection of branch tip positions.

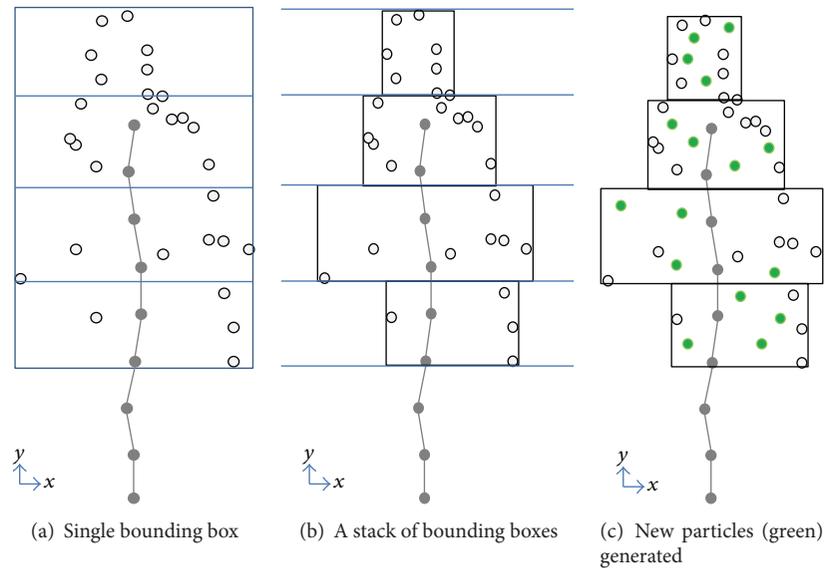


FIGURE 5: New particles are added in a vertical stack of bounding boxes.

TABLE 1: Forces: gravity, shape-format, and dominant wind.

Force	Direction	Magnitude
Gravity	$(0, -1, 0)$	$ S_p - S_t $
Shape-format	$\overrightarrow{S_p - S_r}$	$\alpha \left(\frac{S_{py}^1 + \dots + S_{py}^n}{n} \right)$
Wind	F_w	$ F_w $

S_p : position of a particle; S_{py} : position of a particle in y direction; S_t : position of the nearest trunk point; S_r : position of the root attractor point; F_w : wind measured from motion capture setup; α : scaling weight in range of $[0, 1]$; n : number of particles.

tries to guide particles flow to follow the growth pattern of the original tree. Simulating different growth patterns requires different definitions of the shape-format force. In this research, we provide a simple example of shape-format definition. The shape-format force, shown in Table 1, guides

the particle flow process by height and depth of a tree crown. The direction of the force is pointing to root attractor point S_r from particle location S_p . The magnitude of the force is the average height of all the particles with a weight parameter α . The higher the center of all the particles, the stronger the shape-format force is pointing to the root attractor point S_r . This force is a precomputed global force, which is a constant for all the particles. The direction of the force ensures that a particle finally merges to trunk nodes and therefore all the branches grow towards the trunk.

The dominant wind direction is recorded from the motion capture setup, as described in Section 3. Wind force is a special case of external force acting on tree's branching shapes and structures. While doing motion capture, we record the location of the electrical fan. Because we only use one fan to create wind, that is the only source of explicit external force. Alternatively, the dominant wind direction can be inferred from tree movements recorded in motion captured

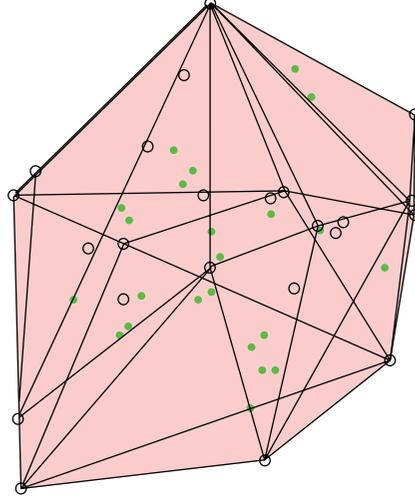


FIGURE 6: A convex hull for all the particles. Particles in black color are from motion capture and particles in green color are randomly generated inside the convex hull.

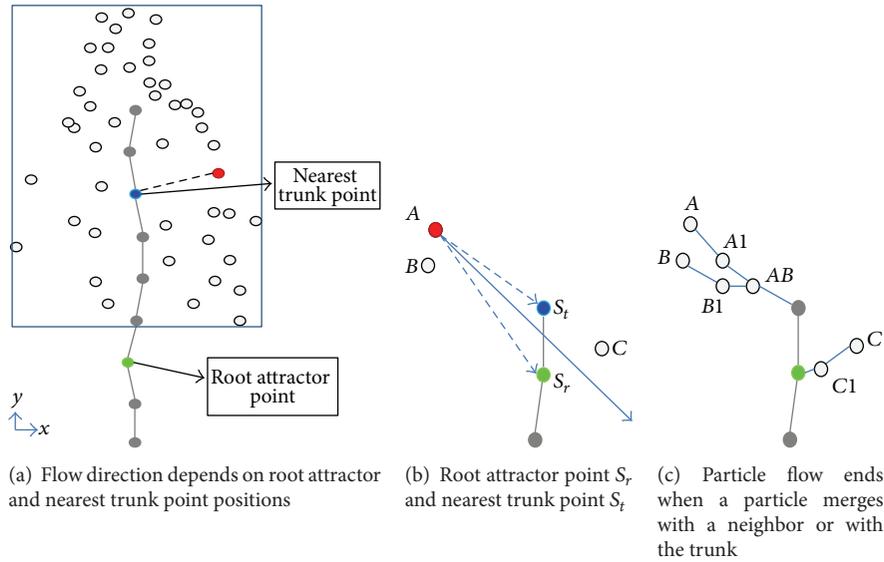


FIGURE 7: A simple particle flow results in crown branching structure.

data. Statistic methods, such as principal component analysis (PCA), can estimate the dominant wind direction.

The flow of particles starts at branch tips. Some particles merge in the flow process while others eventually reach and merge to the trunk. At each time step, we compute step size and direction of a particle. The step size L is

$$L = \beta * \|S_p, S_t\|, \quad (1)$$

where β is a tunable parameter in range of $[0.1, 0.5]$ for most of our tree models. The step direction combines all the three forces. Figure 7(b) shows the computation of the direction for particle i , which is shown as particle A in Figure 7(b). Each force has a weighting parameter ω which tunes the relative

importance of each vector and provides flexibility in creating branching shapes. The particle step direction \vec{D}_i is given by

$$\vec{D}_i = \frac{\omega_1 * \vec{G}_i + \omega_2 * \vec{S}_i + \omega_3 * \vec{W}_i}{3}, \quad (2)$$

where ω is weight of direction, \vec{G}_i is direction of gravity, \vec{S}_i is direction of shape-format, and \vec{W}_i is wind direction for particle i where $i \in [1, \dots, n]$.

Using step size L and direction \vec{D} , the new particle position $V(m)$ is given by

$$V(m) = V(m-1) + L * \vec{D}_i, \quad (3)$$

where m is current time step and $V(m-1)$ is particle position at the last time step.

At each time step, after updating all the particle positions, particles might be merged. When the distance between a pair of particles is less than a predefined merging threshold, those particles are combined. When the distance between a particle and a trunk point falls below a predefined merging threshold, that particle is merged with the trunk. In Figure 7(c), we demonstrate the paths of particle flow. Particles *A*, *B*, and *C* are moved using the step and direction. Particles *A* and *B* are merged at point *AB* and finally merged to the trunk. Particle *C* merged to the trunk point after two time steps of movement.

6. Adding Leaves

Tree leaves are visually important to 3D tree models. After the branching structure is generated, leaves will be attached. We use predefined leaf shapes, growth pattern, density, and size. Also, because leaves do not always start growing at the beginning of a branch, we set a parameter called the leaf starting point. This parameter is proportional to a branch length. For example, when the parameter is 0.3, leaf then starts after the point whose distance to the branch starting point is 0.3 times the branch length.

7. Results

Results are given for multiple trees, including maple and pine trees. The maple tree is instrumented with 24 markers placed on the periphery of the crown at branch tips and the pine tree is instrumented with 35 markers also located on branch tips. We collect the stationary locations of these markers for about 20 seconds at the capture rate of 100 frames/sec.

Figure 8(a) shows all the recorded locations as red dots for a single frame and Figure 8(b) shows averaged locations from each cluster of marker positions for clusters in which the number of frames with a position in that cluster is 1/4 of the total frame count. The data is recorded when the tree is stationary. Notice that a point in the blue box in Figure 8(a) is identified as noise and removed by the clustering algorithm. For the maple tree, after clustering and averaging only 24 markers remain and this matches the actual number of markers placed on the tree.

Although initial marker positions are collected before wind is applied when the tree is stationary, the data contain a small amount of noise, which can be removed to create a single initial marker position. In Figure 9, we show marker positions over time for two markers during the stationary phase. The horizontal axis shows time while the vertical axis shows a marker location in 3D space. In the first image, movement in each direction spans 10^{-3} meters. In addition, in the second image there is apparently a small periodic motion for the stationary branch tip. The range of motion is also within 10^{-3} meters. In both cases, averaging removes this small motion and estimates initial marker positions based on the average rather than a single position in a single frame.

A vertical stack of bounding boxes is a better approximation for the crown volume than a single bounding box and results in better crown shapes. Each bounding box contains

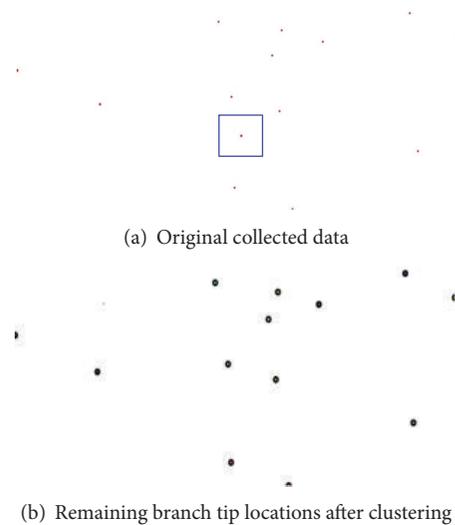


FIGURE 8: Clustering removes noise that leads to spurious marker positions.

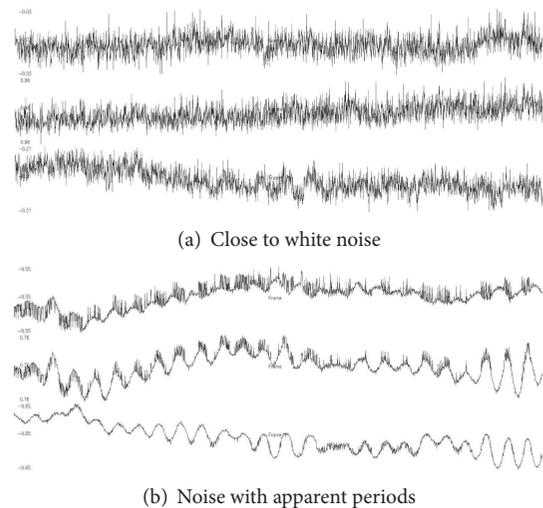


FIGURE 9: Marker positions over time contain a small amount of spurious motion which is removed by averaging.

branch tips and additional branch tips are added to each box. Figures 10(a) and 10(b) illustrate the difference between tree crowns created with and without a stack of bounding boxes for a pine tree. Random particles placed uniformly within a single bounding box result in a cube shaped tree. Placing particles in a vertical stack of bounding boxes better approximates the original crown shape. Future work might include investigating nonuniform distributions of randomly inserted points instead of using bounding boxes.

In Figures 10(c) and 10(d), we demonstrate the difference of tree shapes using stacked box bounding volume and convex hull bounding volume. The bounding box approach provides a looser bounding condition and allows more random factors in the final tree shape. The convex hull approach more closely approximates the original tree crown shape.

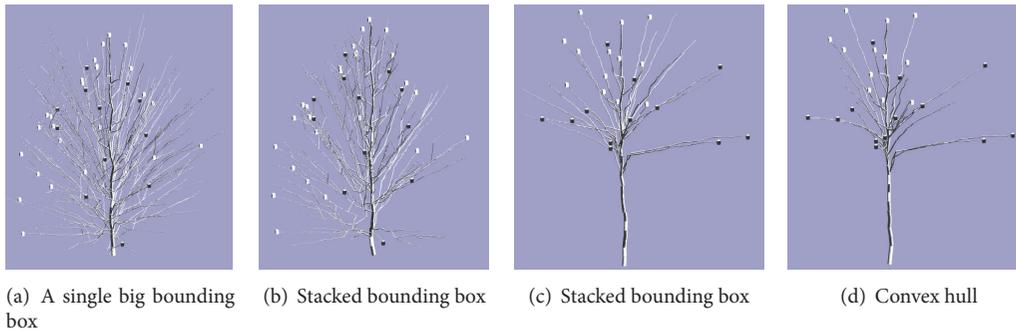


FIGURE 10: Bounding volumes affect tree shape.



FIGURE 11: Pine tree model. 3D tree models created from branch tip positions are similar to the shapes of the trees from which branch tip positions were collected.



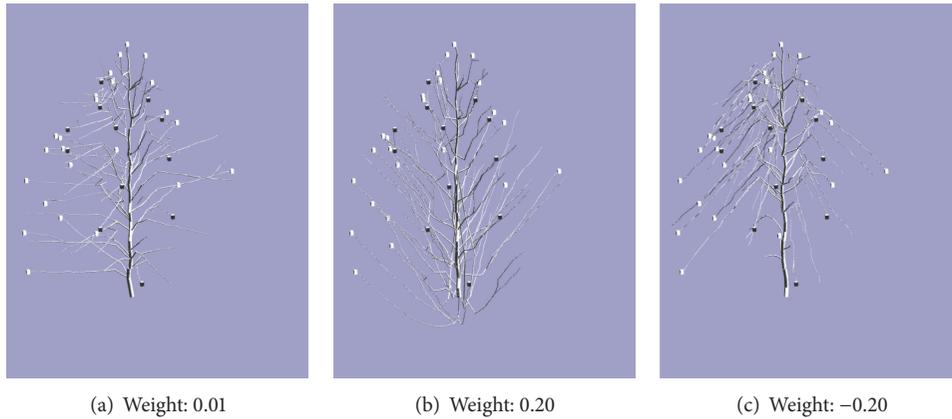
FIGURE 12: Different branching structures under different shape-format forces.

Particle flow starting from branch tips using our simplified algorithm results in tree crown shapes that mimic the shape of the tree from which data was collected. In Figure 11, we show the results from reconstructing a pine tree.

Besides replaying the original tree shape, our approach has flexibility to produce different tree shapes based on the same set of motion capture data. Three forces with their scales create particles' moving paths, which represents branching structures. Figure 12 demonstrates that shape-format force sets the global attracting trunk node, which controls the converging direction of particle flow. The resulting tree models display the tree's growth style in terms of being excurrent and decurrent.

Figure 13 shows the impact of gravity on the tree's shapes with various values of the weighting factor. When the factor is set to be negative, we create a special willow-like tree shape.

Figure 14 shows tree models with different weighting factors of wind force. A bigger value of the factor produces branches bending more toward the wind direction. Notice that the differences among these tree models are small, especially when compared to the influence from the other two forces. This is because with a global wind direction, the value of force is set to be much smaller than the other two forces. Otherwise, when wind force dominates the direction of particle movement, the particles might not be able to merge to a tree's trunk and violate the natural tree's shape.



(a) Weight: 0.01

(b) Weight: 0.20

(c) Weight: -0.20

FIGURE 13: Tree shapes with different weight factors for gravity.



(a) Weight: 0.01

(b) Weight: 0.07

(c) Weight: 0.15

FIGURE 14: Tree shapes with different weight factors for wind force.

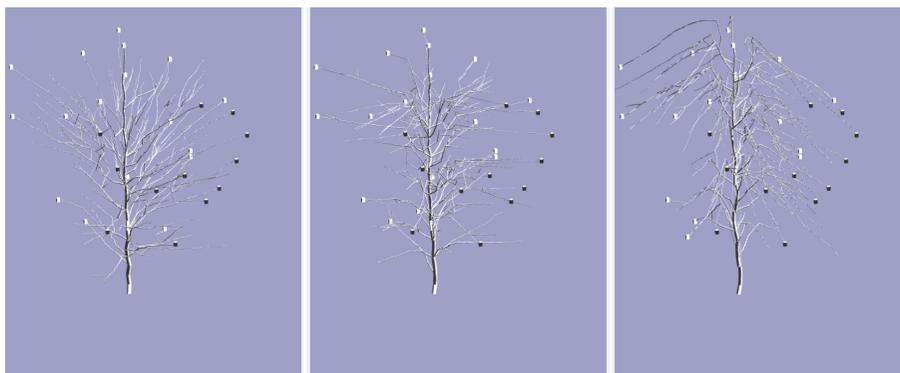


FIGURE 15: Various tree shapes.

In Figure 15, we generate 3D tree models with different parameters for particle flow with the same set of motion captured data. These results demonstrate that our approach produces visually plausible tree shapes, which is scalable for more extended shapes by tweaking parameters of the three forces.

8. Discussion and Future Work

Placing retroreflective markers on branch tips evenly spaced throughout the crown on trees located in a passive optical motion capture arena results in data that can be used to reconstruct tree shape and which may be usable for replaying

branch motion. This can be done using a particle flow system starting from recorded branch tip positions supplemented with additional random branch tip positions within bounding space. We bound the space using a stack of boxes or a convex hull. The procedure of particle flow involves three main forces. With various parameter values of forces, we generate different shapes of trees. A new data collection process designed for trees may extend the use of motion capture to include trees and eventually other networks of nonrigid bodies.

Future work includes extending the process to large trees outdoors as well as improving methods for animating the resulting tree models using the motion capture data. We have reconstructed an approximate tree crown branching structure. Replaying the captured motion data will require care to ensure that the motion of the approximate branching structure does not include uncorrelated motion for branch tips which share a common parent.

References

- [1] A. Lindenmayer, "Mathematical models for cellular interaction in development—parts I and II," *Journal of Theoretical Biology*, vol. 18, pp. 280–315, 1968.
- [2] B. Lintermann and O. Deussen, "Interactive modeling of plants," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 56–65, 1999.
- [3] P. Prusinkiewicz, L. Mündermann, R. Karwowski, and B. Lane, "The use of positional information in the modeling of plants," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 289–300, ACM, August 2001.
- [4] J. Weber and J. Penn, "Creation and rendering of realistic trees," in *Proceedings of the 22nd ACM Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, pp. 119–126, Los Angeles, Calif, USA, August 1995.
- [5] A. Reche-Martinez, I. Martin, and G. Drettakis, "Volumetric reconstruction and interactive rendering of trees from photographs," in *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '04)*, pp. 720–727, ACM, Los Angeles, Calif, USA, August 2004.
- [6] B. Neubert, T. Franken, and O. Deussen, "Approximate image-based tree-modeling using particle flows," in *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '07)*, ACM, San Diego, Calif, USA, August 2007.
- [7] P. Tan, G. Zeng, J. Wang, S. B. Kang, and L. Quan, "Image-based tree modeling," in *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '07)*, ACM, San Diego, Calif, USA, August 2007.
- [8] C. Li, O. Deussen, Y.-Z. Song, P. Willis, and P. Hall, "Modeling and generating moving trees from video," in *Proceedings of the SIGGRAPH Asia Conference (SA '11)*, pp. 1–12, ACM, Hong Kong, December 2011.
- [9] J. Diener, L. Reveret, and E. Fiume, "Hierarchical retargetting of 2D motion fields to the animation of 3D plant models," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '06)*, pp. 187–195, Eurographics Association, Aire-la-Ville, Switzerland, 2006.
- [10] H. Xu, N. Gossett, and B. Chen, "Knowledge and heuristic-based modeling of laser-scanned trees," *ACM Transactions on Graphics*, vol. 26, no. 4, 2007.
- [11] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Transactions on Graphics*, vol. 29, no. 6, article 151, pp. 1–8, 2010.
- [12] Y. Livny, S. Pirk, Z. Cheng et al., "Texture-lobes for tree modelling," *ACM Transactions on Graphics*, vol. 30, no. 4, article 53, pp. 1–10, 2011.
- [13] W. T. Reeves, "Particle systems—a technique for modeling a class of fuzzy objects," *ACM Transactions on Graphics*, vol. 2, pp. 359–376, 1983.
- [14] A. Runions, B. Lane, and P. Prusinkiewicz, "Modeling trees with a space colonization algorithm," in *Proceedings of the Eurographics Workshop on Natural Phenomena*, 2007.
- [15] W. Palubicki, K. Horel, S. Longay et al., "Self-organizing tree models for image synthesis," in *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '09)*, pp. 1–10, ACM, 2009.
- [16] J. Long and M. D. Jones, "3D tree modeling using motion capture," in *Proceedings of the 4th IEEE International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications (PMA '12)*, pp. 242–249, 2012.
- [17] H. Lou and J. Chai, "Example-based human motion denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 5, pp. 870–879, 2010.
- [18] S. Rajko and G. Qian, "Real-time automatic kinematic model building for optical motion capture using a markov random field," in *Proceedings of the IEEE International Conference on Human-Computer Interaction (HCI '07)*, pp. 69–78, Springer, Beijing, China, 2007.
- [19] G. Wen, Z. Wang, S. Xia, and D. Zhu, "From motion capture data to character animation," in *Proceedings of the 13th ACM Symposium on Virtual Reality Software and Technology (VRST '06)*, pp. 165–168, ACM, Limassol, Cyprus, November 2006.
- [20] A. G. Kirk, J. F. O'Brien, and D. A. Forsyth, "Skeletal parameter estimation from optical motion capture data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 782–788, June 2005.
- [21] W.-C. Ma, A. Jones, J.-Y. Chiang et al., "Facial performance synthesis using deformation-driven polynomial displacement maps," *ACM Transactions on Graphics*, vol. 27, no. 5, article 121, pp. 1–10, 2008.
- [22] E. Sifakis, I. Neverov, and R. Fedkiw, "Automatic determination of facial muscle activations from sparse motion capture marker data," *ACM Transactions on Graphics*, vol. 24, pp. 417–425, 2005.
- [23] J. Long, C. Reimschuessel, O. Britton, A. Hall, and M. Jones, "Motion capture for a natural tree in the wind," in *Proceedings of the 3rd International Conference on Motion in Games (MIG '10)*, pp. 158–169, Springer, Utrecht, The Netherlands, 2010.
- [24] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions on Graphics*, vol. 13, no. 1, pp. 43–72, 1994.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

