Visual 3D Reconstruction and Dynamic Simulation of Fruit Trees for Robotic Manipulation

Francisco Yandun, Abhisesh Silwal, George Kantor
Field Robotics Center, Carnegie Mellon University
5000 Forbes Avenue
fyandun@andrew.cmu.edu, asilwal@andrew.cmu.edu, gkantor@andrew.cmu.edu

Abstract

Modern agriculture is facing a series of challenges to adopt new technologies to improve sustainability, profitability and resilience. One of them is the use of robotic applications to assist or even replace manual workers for the complex task of interaction with the vegetation. For example, harvesting and pruning are tasks that need certain dexterity to not only make the cuts, but also to move branches or foliage in the canopy to reach hidden objects or locations. For such capability, first the robot should be able to perceive the vegetation and estimate the dynamics for the interaction. This work mainly focuses on the perception problem, aiming to digitize commercial tree fruit canopies and estimating how it moves when force is applied to the branches. We studied the suitability of two known algorithms, viz. the space colonization and the Laplace based contraction algorithms, to build a geometric model of the tree using point cloud data from stereo cameras. Such model is then used to estimate the dynamics of the tree, by considering the branches as links articulated by spring-damper joints. The geometric model was evaluated for topological and morphological correctness by comparing it with the ground truth, obtaining better results with the Laplace based contraction algorithm. Furthermore, results of the dynamics estimation showed that by adjusting the parameters for the spring-damper model, the motion prediction is promising, with a maximum mean squared error of 0.073m in the tracking of the movement of the branches.

1. Introduction

The research and development of automated or semi-automated robots to gather information about the vegetation status has the potential to address many of the challenges that modern agriculture entails [14]. However, there are numerous other applications that require physical interaction with the vegetation using robotic manipulators, and these challenges remain largely unsolved. For example, low availability of skilled human workforce and the hard working conditions are serious issues in pruning and harvesting that affect not only the productivity but also causes ergonomic injuries from repetitive tasks [20, 5]. For a robot to perform or support these tasks robustly, we hypothesize that the ability to build a model of the tree structure and its kinematics and dynamics is necessary to manipulate the plants. This leads to the ability to estimate how the structure will respond during and after the interaction. This feature would be highly valuable when obstacles need to be pushed away to reach objects of interest that may be (partially) hidden in the canopy.

In this work, the structural modeling of a tree refers to the use of images to build skeletal representation by detecting certain points of interest (i.e., positions of the root and leaves) and geometric characteristics including width of the branches and their position in the space. To this aim, techniques including L-systems, space colonization, graph-based modeling, generalized rotational symmetry axis (ROSA), among others have proven to get realistic models in different computer graphic applications [34]. However, using the generated models in applications beyond simulations and computer games is still a remaining gap, specially for agricultural robotic applications.

If the skeletons produced by any of the mentioned approaches are represented as a series of joints and links, they could be used to estimate the effects of moving one branch in the overall tree structure. Precisely, the kinematic and dynamic modeling serves this purpose, providing a way to estimate not only the motion of the tree when interacting with the robot, but also in the presence of other perturbations such as wind. Additionally, with a complete and accurate structural/dynamic model, the robot may be able to judiciously push away branches or obstacles within the canopy to deal with the occlusion problem. While any dynamic model for an articulated body could serve for this purpose, there is an additional and important characteristic to be taken into account: the spring-damper behaviour

of branches in a tree [12]. Then, it is natural to think that branches in a tree can be modeled as a series of elastic rods with three rotational degrees of freedom, whose dynamics will be characterized by the relation between the angular position of the branch, its speed and the spring and damping coefficients [2].

Given the previous context, we present and evaluate a methodology for digitization of fruit trees using visual or range sensing to generate a three dimensional model using the space colonization algorithm and the Laplace based contraction method, respectively. Such geometrical model is subsequently used to estimate the dynamics of the tree represented as a series of linked rigid bodies, where the stiffness of each joint is function of spring and damping constants that can be identified for different types of trees.

We do not aim to develop a new tree modeling methodology, but the contribution of our work is to use and further develop computer graphic methods to give a step beyond and apply them to represent fruit trees in a way that would be useful to estimate their movement when interacting with a robot in real world agricultural applications. Furthermore, we provide a practical methodology to combine the geometric and dynamic models and to quantitatively evaluate the fidelity of such representations in real fruit trees.

This paper is organized as follows: Section 2 reviews the related work about structural and dynamic modeling of trees for diverse applications. Section 3 describes the hardware employed, as well as the methodology followed. In Section 4, we present the results of evaluating the employed approach in synthetic and real datasets, along with laboratory tests. Finally, the conclusions of our work are described in Section 5.

2. Related work

Realistic tree modeling aims to faithfully and accurately represent two aspects of the tree structure: its visual appearance and movement. The first deals with the generation of the branching configuration and shape of a tree (which changes among the different species of trees found in nature). The second focuses on how the generated model would react in presence of perturbations or when interacting with other objects. Both have been widely studied in the context of botany to gain understanding about the grow patterns [24, 25], and computer graphics to design virtual environments for plant simulation [18]. Procedural approaches such as Lindenmayer systems (L-systems) and its variations have proven to produce virtual tree geometries that resemble real structures of diverse types of species [26, 7]. Other methods employ the concept of competition for resources (e.g., light or space) to build the structure in a similar way as nature does in plenty of cases. For example, the space colonization algorithm [30] uses a parameterized way to control the structure growth based on the competition of the branches to reach the position of the leaves.

Plenty of applications of L-systems and space colonization have been used to synthesize virtual trees with similar characteristics to real ones, but not to reconstruct real structures using sensors that observe the plants. However, there are works that reported the use of imagery to extract parameters for developing L-systems rules that allows to grow a model that resembles the scanned tree [33, 8]. Other image-based methods propose to build three dimensional tree models based on sketches or silhouettes of the tree canopy [38, 16]. Although not designed specifically for tree structures, recent works using statistical inference and deep learning have shown promising results [15, 23].

As an alternative to images, data from 3D laser scanners have also been used to build virtual models of trees. The point cloud generated is usually employed to obtain a skeleton representation of the structure as a hierarchical graph. For example, a branch structure graph was constructed and refined by a series of optimizations driven by certain assumptions of the tree geometry in [17]. Similarly, an optimization approach was used in [36] to estimate a set of parameters for their model that better resembles the input tree under shape, structural and geometric metrics. In fact, geometric properties of a point cloud are also commonly used to get skeletonized structures of different objects, as reported in [11]. For tree modeling applications in specific, the improvement of the centerness and topology of skeletons produced by an octree method was proposed in [9]. They re-center the structure based on the symmetry axes of cylindrical primitives fitted to the branches. The topology is further refined using clustering algorithms and spline interpolation. In the context of agricultural environments, 3D point clouds were used in [4] for generating a model to later estimate the length and width of branches of fruit trees. The same author in [3] also provided a brief summary of the applications of the skeletonization of trees for agriculture and forestry. Additionally, the use of simulated range sensor readings from artificially generated fruit trees, was studied in [19].

Complementary to visual modeling of trees, the study of the mechanics of its movement have also been studied. Trees are usually represented as articulated bodies with a spring-damper behaviour on its branches, whose dynamics are characterized as multiple mass dampers [35, 12]. The movement is then propagated from the collision point to the rest of the tree, as in [32, 27] and the references therein. Furthermore, an interesting feature of these approaches is that they can be parameterized using the stiffness and damping coefficients, which allowed to model various structures in [39]. As an alternative, trees have also been modeled as series of rods interconnected, as presented in [10].

Finally, it is important to note that the majority of methods for tree modeling discussed here are not originally

quantitatively compared to the real counterparts. This observation is coherent with the fact that most of them are computer graphics applications whose aim is to create virtual environments that visually render real looking trees. However, when the aim is to manipulate the tree, the accuracy of the model obtained is an important feature to be evaluated.

3. Materials and Methods

The proposed approach comprises two stages: the three dimensional modeling of the tree and its dynamical simulation. For the first stage, we implemented two methods from the state of the art: the space colonization algorithm [30] and the Laplacian based contraction [6]. We consider that both are purely visual since the sensor employed to get the real data is a camera system, as will be described in the following Section. These two algorithms were chosen mainly for the following three reasons. First, they are well known methods to skeletonize and reconstruct branching structures [28, 1, 22]. Additionally, they have proven effectiveness in various scenarios and datasets [8]. Finally, we consider that they provide a complementary way to model the tree structure in the case of a tree with foliage, as will be discussed in Section 4.1.

The dynamic simulation was required to be flexible in the sense of allowing joints with elastic stiffness with 3 degrees of freedom. Furthermore, the computation of the movement required a robust and efficient method so later it can be used for a robot to estimate it movement or actuate on the tree. While known physics engines (e.g., bullet, open dynamics engine, PhysX) provide these two characteristics, they seemed inadequate to explicitly model and tune the elastic stiffness in each degree of freedom. For these reasons, we implemented the model described in [27], which is a computer graphics application for botanical tree animation that suits better for the purposes of this work.

Figure 1 summarizes the pipeline we followed to obtain the geometric structure of a fruit tree and estimate its dynamics.

3.1. Development Datasets and Hardware

We used three types of datasets to evaluate the quality of the 3D reconstruction and the feasibility of the dynamic modeling for real trees. The development-artificial dataset consisted in 4 freely available skeletons (with their respective point clouds) of different shaped and sized trees, provided in [9]. The skeletons were considered as ground truth for all of them. The development-real dataset comprised a set of 14 point clouds from real grape vines, acquired in field with a semi-autonomous platform equipped with dual stereo cameras and a gantry system (Fig.2). We basically used the Iterative Closest Point (ICP) algorithm and rigid transformations to align the point clouds produced by the top and bottom cameras. As this paper mainly focus on the visual 3D reconstruction and dynamic simulation of tree fruits, the design of the image acquisition system and the platform are targeted for another independent publication.

The third dataset corresponded to laboratory tests with a dormant vine, which was mainly used to evaluate the practical applicability of the movement estimation in real scenarios.

3.2. Image-based approach - Space colonization algorithm

The Space Colonization Algorithm (SCA) is an iterative method to build tree structures based on the concept of competition for space. It takes as input N attraction points (which can be the leaves or buds of the tree) and the tree root or other pivot points acting as prior nodes (joints). Then, the algorithm builds the branching structure of the tree by attaching new links based on their distance and relative position to the nearest attraction point(s). A deeper description of the algorithm, as well as some interesting modifications can be found in the original work [30].

The natural interpretation of the attraction points from a practical point of view is to consider them as leaves. However, we employed other possibility: detect the buds of the branches and use them as attraction points. In this work, the detection procedure relied on a Faster-RCNN, which is a very popular deep neural architecture for object recognition [29]. The consecutive incoming images from top and bottom stereo cameras are sent through the object detector trained to detect dormant buds in 2D images. Using the optimized transformation of the registered point cloud produced by the ICP algorithm (see Sec. 3.1), the dormant buds were filtered and merged. In this case, it was necessary to find appropriate values for the growing parameters that force the branches to reach the buds. Furthermore, in some cases (specially for the vines) one additional pivot was given as input for the tree to obtain a structure similar to the real. In this case, the algorithm starts trying to reach only the pivot for later start growing using the other attraction points.

3.3. Point cloud based approach - Laplacian-based contraction

Among the diverse algorithms for tree skeletonization, we employed the Laplacian-based contraction (LBC) [6]. It is a geometric approach that collapses the input point cloud maintaining its global shape. The main idea is to obtain a new point cloud with minimal volume using a repetitive Laplacian smoothing process. At each iteration i, the point cloud P is contracted by finding the set of contracted points

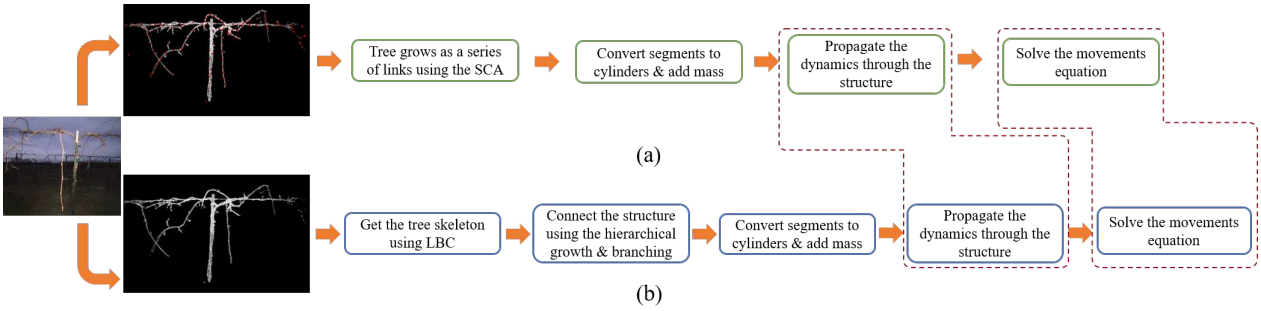


Figure 1: Workflow of the proposed approach to digitize and simulate the movement of fruit trees using A) the space colonization and B) a skeletonization algorithms.

Figure 2: A semi-autonomous platform for image acquisition. The imaging system consists of two stereo cameras attached to a gantry. The stereo cameras acquire seven different top and bottom views of the vine at a regular interval along the gantry.

P' that solve the linear system:

$$\begin{bmatrix} W_L^i L^i \\ W_H^i \end{bmatrix} P'^i = \begin{bmatrix} 0 \\ W_H^i P^i \end{bmatrix} \quad (1)$$

where W_L, W_H are weighting factors to control the contraction and attraction constraints, and L is the Laplacian operator with cotangent weights. For the next step, the new set of points found in the current iteration, P'^i, is assigned to P^{i+1} while the weights and the Laplacian operator are updated to account for the "new shape" of P^{i+1}. This process repeats until a convergence test is passed.

When carefully designing the weighting factors, this overall approach provides efficiency and stability when contracting point clouds. As the resulting set of points is a thin structure that approximates the skeleton of the original point cloud, it is further necessary to connect the structure ensuring to faithfully represent the topology of the original structure (a tree in our case).

3.3.1 Topological connection

Various methods have been proposed to convert a point cloud to a skeleton [13, 31]. However, we used a custom method to connect the points obtained from the LBC that aims to resemble the characteristic topological structure of a tree. We developed this method is specific to account for the hierarchical pattern that the dynamic simulation requires. This is, starting from the root, the subsequent links require a series of parent-child connections that preserves the tree shape to properly propagate the movement when a force is applied to any branch.

Firstly, the points are organized in ascending order according to their z coordinates, obtaining a point cloud $P = \{p_i\}$. Denoting a node in the interconnected graph as $g_i = \{p_i, \alpha_k^i\}$, where α_k^i is its parent node with index k, we use the procedure summarized in Algorithm 1.

The main idea is to join the children nodes with their parents based on its Euclidean distance. If a node has multiple parents during the connection, the one with less distance adopts that node as child. However, there may be cases when this distance test is not conclusive (lines 12 and 17). In this circumstance we use the observation that the connection direction is influenced by the spatial distribution of the points along the branches or the trunk. The neighbors oriented search (lines 13 and 18) is used to implement this idea as follows: we first calculate the directional vectors from both disputing parents to the current node. Two imaginary cylinders oriented in these directions, with fixed radius ρ and length L_{cyl} depending on the dimensions of the tree are then built, as shown in the third row of Fig 3c. The points lying inside each cylinder are counted and the one with majority is chosen. Finally, the parent associated to that cylinder keeps the child.

The overall procedure, is summarized in Fig. 3 as fol-

Algorithm 1 Topological tree connection

Make $G'(g_i) = \{p_i, \emptyset\} \quad \forall p_i \in P$
while nodes without a parent exist **do**
  Increase index i
  Find nodes $g_j : |p_i - p_j| < r$
  while g_j is not found **do** increase r
  For all $g_j : \alpha^j = \emptyset$, connect p_j to p_i and make $g_j = \left\{p_j, \alpha_i^j\right\}$.
  if $\exists\, g'_j = g_j : \alpha_k^j \neq \emptyset$ **then**
    for all g'_j **do**:
      $d_{curr} = |p_j - p_i|, \quad d_{prev} = |p_j - \alpha_k^i|$
      if $d_{curr} \leq d_{prev}$ **then**
        if $\frac{d_{curr}}{d_{prev}} < d_{thres}$ **then**
          Change the parent $\alpha_k^j = \alpha_i^j$
        else
          Do neighbor oriented search
      else
        if $\frac{d_{prev}}{d_{curr}} < d_{thres}$ **then**
          Keep the previous parent α_k^j
        else
          Do neighbor oriented search
  Make $G = G'$
  Remove g_i from G'.

lows: the first row shows a simple case when a multi parent node is found. The parent is assigned based on the shorter distance to the disputing node. Similarly, the second row shows a multi-parent node in a branching point, where the dispute is solved based on the distance. The third row shows the more complex case where the distance test is not conclusive and the neighbor oriented search is performed. As the orange cylinder has more points inside the original parent is finally chosen. This process is repeated until all the points have a parent assigned. It is noteworthy that the presented method requires three design parameters: the radius to make the initial search r, the ratio to compare the distances between disputing parents d_{thres} and the radius of the oriented cylinders ρ. While its length can be also a parameter we choose to make it function of the tree height to avoid an over-parametrization of the method.

3.4. Dynamic modeling

The connected structure from the SC and LBC algorithms are just skeletal representations of the tree. To convert them into 3D structures capable to move and interact with other objects, we assume that each link has a cylindrical shape, whose length is known, but the radius changes depending of the tree structure. This change was modeled making the radius a linear function of the number of descendants of each joint (this accounts for the complete lineage,

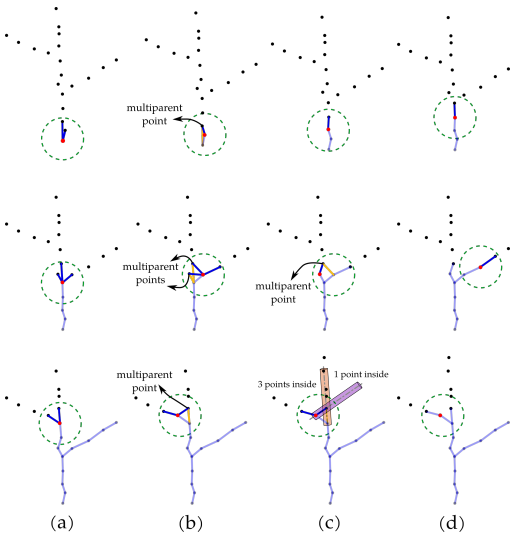


Figure 3: Topological connection applied to three different scenarios, increasing in complexity. The red dot depicts the current node g_i, the blue lines are the current connections based on the distance, and the yellow lines show the conflicting connection with previous parents.

not only the current children). The inertial properties of each link were then included by estimating the mass using the volume of each cylinder and the density of the wood for the tree under study. The movement of the tree was calculated by using the four steps originally proposed in [27] and described as follows:

- External force computation: For each rigid body (cylinder), external forces and torques due to gravity, wind fields or other sources are calculated.
- Composite body update: The inertial characteristics of a link of the tree are at some extent affected by the others. Therefore, the total external forces and torques, mass, and inertia tensor (in world reference frame) of each rigid body are calculated in base of the branching structure of its children and its own. These calculations are backpropagated from the outer links to the root in order to update all the tree structure.
- Analytic spring evolution: The joint space configuration is then calculated by solving the equation:

$$\tilde{I}\ddot{\boldsymbol{\theta}} + \beta K\dot{\boldsymbol{\theta}} + K\theta = \tilde{\boldsymbol{\tau}} \tag{2}$$

where $\boldsymbol{\theta}$ is the joint position, $\tilde{I}$ is the local composite inertial tensor, K is a diagonal 3×3 matrix whose elements correspond to the rotational stiffness of the joint, βK accounts for the damping coefficients and $\tilde{\boldsymbol{\tau}}$ is the local composite external torque.

- Rigid body state update: The state vector of each link $[m_i, \mathbf{x_i}, \mathbf{v_i}, \mathbf{a_i}, I_i, R_i, \boldsymbol{\omega_i}, \boldsymbol{\alpha_i}]$ is expressed in maximal coordinates and includes its composite mass, position, velocity, acceleration, composite inertia, rotation matrix, angular velocity and angular acceleration, respectively. Once the solution of Eqn.(2) is found for all the links, their state vectors are updated, starting from the root.

While for a virtual environment simulation, tuning K to obtain different spring behaviors of the joints would be enough, the modeling real fruit trees requires a more analytical approach. For that reason, we used the definition of K described in [21]. In this work, each of the elements of K is related with the area moment of inertia of the link, and structural constants (available for various materials) such as the Young modulus and the Poisson's ratio. Thus, we have an a prior knowledge of K, which can be used as an initial condition when performing identification tests in specific trees, for example.

4. Evaluation metrics and results

The validation of the tree geometric and dynamic models were twofold. First, we used the two development datasets to evaluate the geometric modeling. In these datasets, we mainly evaluated the topological and morphological correctness of the obtained model with respect to the input ground truth and the point clouds.

To assess the topological correctness, we quantified the matching between branching and termination points (BP and TP), among the ground truth and our model. The BP and TP ground truth was manually obtained by visual inspection of the development-synthetic skeletons. In the generated model, a BP is a joint forming a vertex greater or equal to 3^o. On the other hand, a TP is a joint with no children. To find the BP and TP in our model that matches the ground truth, we employed the Euclidean distance. Since the trees in the dataset had different sizes, we evaluated two thresholds to define the matching: a fixed value Δd_1 and another depending on the tree height Δd_2. Then, accuracy and recall metrics were calculated as,

$$\text{Accuracy} = \frac{MP}{TP_{model}}, \quad \text{Recall} = \frac{MP}{TP_{GT}}$$

where MP is the number of matched points, TP_{model} and TP_{GT} are the number of points in the model and the GT, respectively.

The morphological correctness of the model was evaluated using the Hausdorff distance, which allowed us to measure the geometric difference between two point clouds or meshes [37]. To generate the point cloud of the model, we uniformly sampled a set of points around the structure. Then, the Cloud Compare software was employed to obtain the Hausdorff distance. Both topological and morphological correctness were used in complement for the development-synthetic dataset. However, we only assessed the morphological correctness in the development-real dataset since it only consisted in point clouds, which made it impractical to accurately find the ground truth BP and TP.

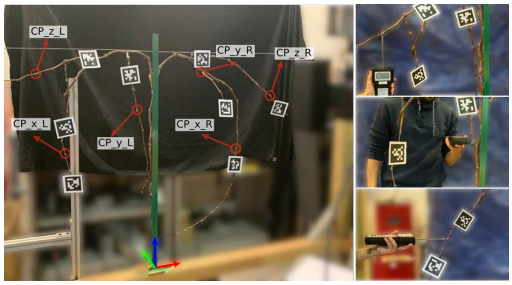


Figure 4: Experimental setup to evaluate the dynamic model. The contact points are encoded according to its location and the axis on which the force was applied. The right column shows how the sensor was employed to measure the force magnitude for each axis.

To evaluate the dynamic modeling, we used laboratory tests on a dormant vine. We placed several fiducial tags in the tree structure to track its movement using cameras in a stereo pair, as shown in Fig 4. Subsequently, various forces were applied along each axis x, y and z, and the displacement of the markers saved. The contact point locations for each case were marked in the real structure and then identified in the virtual model. Additionally, the magnitude of the applied force in each trial was measured using a force gauge with 49N capacity and $\pm 0.4\%$, 1 digit accuracy. We then calculated the mean squared error (MSE) between the measured final tag positions and those predicted with the dynamic model, as described by Eqn. 3.

$$MSE_j = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x_{i,j}} - \hat{\mathbf{x_{i,j}}})^2 \tag{3}$$

where N is the number of markers, $\mathbf{x_{i,j}}$ and $\hat{\mathbf{x_{i,j}}}$ are the measured and estimated positions of the tag i for the trial j, respectively.

4.1. Space colonization algorithm

Results of the evaluation of the topological correctness in the development synthetic dataset are summarized in Table 1. It can be noted that the SCA fails to accurately match the BP and TP with the ground truth, specially for Δd_1. This outcome is at some extent expected since in the model, the tree growth with no other constraints than the positions of the buds, acting as attraction points. Additionally, it has

Table 1: Summary of topological correctness performance for the SCA.

		Tree 1		Tree 2		Tree 3		Tree 4	
		Δd_1	Δd_2	Δd_1	Δd_2	Δd_1	Δd_2	Δd_1	Δd_2
Bifurcation Points	Accuracy	0.231	0.231	0.015	0.162	0.034	0.101	0.002	0.038
	Recall	0.286	0.286	0.091	1.000	0.333	1.000	0.052	0.974
Termination Points	Accuracy	0.429	0.429	0.114	0.343	0.140	0.207	0.016	0.045
	Recall	0.273	0.273	0.333	1.000	0.680	1.000	0.328	0.916

$\Delta d_1 = 0.25$ m
$\Delta d_2 = 5\%$ of tree height

to be noted that recall is consistently better than accuracy, which indicates that the model was able to match the BP and TP of the ground truth, but it also provided a lot of false detections.

Figure 5 shows the boxplots for the Hausdorff distance for the models built using the SCA for both development datasets. Additionally, right columns of Figure 5 shows examples of two point clouds built based in the SCA model for the synthetic and real datasets. They are colorized according to the Hausdorff distance to the ground truth data. In general, it can be seen that various parts of the generated structures are similar to the ground truth under this metric (blue-colored points), but the distance grows near the branching points, which agrees with the results of Table 1, where the accuracy in the BP and TP matching was low.

Despite of the low performance of the SCA in these tests, the results still show potential considering that trees of both datasets are dormant, with their branching structure in sight. However, the SCA can be useful in trees with leaves, where such structure is hidden beneath the canopy. In this case, the overall system can work in two iterative stages: first, detecting the leaves positions (with any object detection algorithm) to use them along with the SCA to compose an initial "guess" of the branching structure. The second step would require to move the tree to obtain a better perspective of its branching configuration, providing thus a feedback to refine the generated model.

4.2. Laplacian-based contraction

In this case, the accuracy of the BP and TP matching certainly improved, as Table 2 shows. However, the termination points recall is specially comparable (and in some cases worst) to the values obtained with the SCA. This outcome implies that the TP of the tree model agreed better to the ground truth, but it also produced extra spurious TP, producing a decrease in the recall. Making a close insight of the model, these spurious TP were generated by noisy points throughout the branches. To improve this issue we have to explore ways to improve the topological connection or smooth the skeleton produced by the LBC.

The morphological correctness of the model for the LBC algorithm outperformed the SCA, as Fig. 6 shows. It can be seen that the Hausdorff distances are considerable smaller

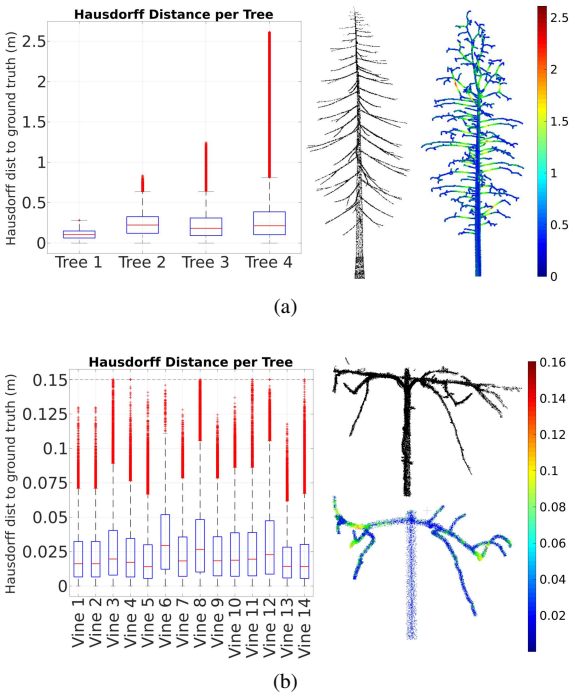


Figure 5: Results for the space colonization algorithm. The left figures show the boxplots for the Housdorff distances between our model and the ground truth for the synthetic and real datasets. The right figures show examples of the input point clouds (black points) along its generated models, colorized according to their Hausdorff distances to the input point cloud, measured in meters.

Table 2: Summary of topological correctness performance for the LBC algorithm.

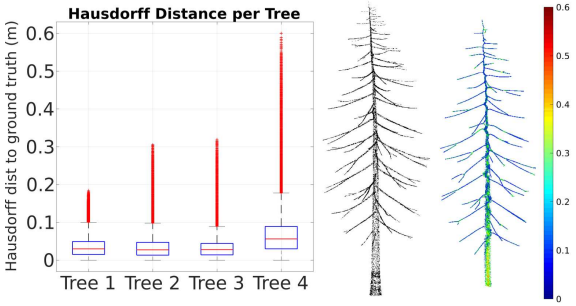
		Tree 1		Tree 2		Tree 3		Tree 4	
		Δd_1	Δd_2	Δd_1	Δd_2	Δd_1	Δd_2	Δd_1	Δd_2
Bifurcation Points	Accuracy	0.293	0.293	0.333	0.458	0.417	0.500	0.063	0.556
	Recall	0.571	0.571	0.727	1.000	0.833	1.000	0.078	0.687
Termination Points	Accuracy	0.455	0.455	0.154	0.923	0.120	0.960	0.014	0.986
	Recall	0.455	0.455	0.167	1.000	0.120	0.960	0.008	0.605

$\Delta d_1 = 0.25$ m
$\Delta d_2 = 5\%$ of tree height

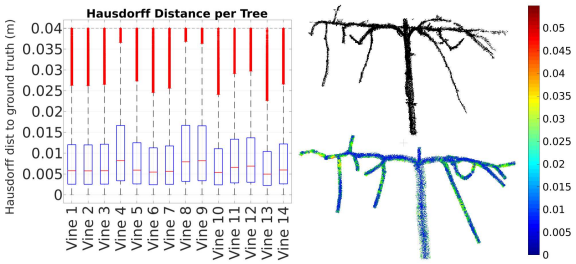
than the ones obtained previously, which indicates that LBC algorithm combined with our topological connection provides an overall reliable representation of the tree morphology. This result can be also verified in the right columns of Fig 6, where two point clouds colorized according to their Hausdorff distance to the ground truth are depicted.

4.3. Tree dynamics

To generate the tree geometry, we employed the LBC algorithm for the laboratory vine. As the aim in these tests was to validate only the dynamics, the input point cloud was manually enhanced to remove any artifact or spurious point



(a)



(b)

Figure 6: Results for the Laplace based contraction and topological connection method. The left figures show the box-plots for the Hausdorff distances between our model and the ground truth for the synthetic and real datasets. The right figures show examples of the input point clouds (black points) along its generated models, colorized according to their Hausdorff distances to the input point cloud, measured in meters.

that may affect the geometric model. Additionally, several prior tests were conducted to adjust the value of the K parameters that correctly account for the observed movement of the tree. For these tests, we obtained 5 values of K that provided the best results in terms of the final tree shape and the motion of the marked points. The requirement of this manual adjustment for different cases reveals two important points of the dynamic model employed. First, with a suitable value of K, the model can properly account for the tree movement in presence of external forces. Accordingly, it is required an identification/learning approach to estimate K such that it generalizes to movement in all directions. Furthermore, the linear mapping of the cylinders radius as function of the number of descendants is not entirely accurate. Instead, the radius of certain segments of the branches can be estimated from the point cloud.

The results of the dynamic model validation tests are summarized in Table 3. Given the tree dimensions in height, length and depth are 0.8m, 1.65m and 20.26m, respectively; we consider that the model correctly accounted for the tree motion, given the remark about the spring stiffness K explained above.

Table 3: Mean squared error of predicted positions for the fiducial markers attached to the tree.

Axis of motion	x		y		z	
Contact Point Id	CP_x_L	CP_x_R	CP_y_R	CP_y_L	CP_z_R	CP_z_L
Trial 1	0.015	0.051	0.069	0.036	0.034	0.030
Trial 2	0.020	0.043	0.073	0.050	0.038	0.044

5. Conclusions

This work presented the geometric and dynamic modeling of fruit trees using computer graphic methods mainly designed to create virtual environments. In contrast to common applications in this field, the digitization of real trees comprises a series of additional challenges that we addressed. First, it is not enough for the model to have a structure similar to a generic tree, it needs to have - approximately- the same branching disposition, shape and dimensions of the real one. We used point clouds from synthetic and real trees (the latter acquired in field) to generate a model using two known methods such as the space colonization and the Laplacian based contraction algorithms. They were subsequently assessed quantitatively in order to test the suitability of each one to account for such characteristics. Results from topological and morphological correctness test shown that the Laplacian based contraction together with our topological connection algorithm performed better in dormant trees. However, the topological correctness tests with the synthetic dataset showed that although it can properly resemble the BP and TP, it is prone to produce small spurious termination points.

Another challenge we addressed when digitizing real trees consisted in obtaining a motion model that accounts for the springy behaviour of the branches. To this aim, we used a dynamic model for rigid bodies articulated by stiff joints. Furthermore, its parametrization allowed a spring-damper interpretation to simulate diverse behaviours for different trees. Considering the input is a good geometrical model (using either SCA, LBC or other algorithm), it is capable to accurately predict the motion of the branches in presence of external forces, given the proper value of the parameters.

The ongoing work is focusing in the correct identification of the K parameter. This is specially critical since the overall accuracy of the motion prediction relies on it. Finally, with a complete geometric and dynamic model of a fruit tree, the final objective is to use it in practice for a robot to intelligently interact with the canopy in activities including automated harvesting or pruning.

References

[1] Oscar Kin Chung Au, Chiew Lan Tai, Hung Kuo Chu, Daniel Cohen-Or, and Tong Yee Lee. Skeleton extraction by mesh

contraction. *ACM Transactions on Graphics*, 27(3):1–10, 2008. 3

[2] Franka Brüchert, Olga Speck, and Hanns Christof Spatz. Oscillations of plants' stems and their damping: Theory and experimentation. *Philosophical Transactions of the Royal Society of London*, 358(1437):1487–1492, 2003. 2

[3] A Bucksch, RC Lindenbergh, and M Menenti. Applications for point cloud skeletonizations in forestry and agriculture. Technical report, 2009. 2

[4] Alexander Bucksch, Roderik Lindenbergh, Massimo Menenti, and Muhammad Z. Rahman. Skeleton-based botanic tree diameter estimation from dense LiDAR data. In *Proceedings SPIE 7460, Lidar Remote Sensing for Environmental Monitoring*, pages 1–11, 2009. 2

[5] Bülent Çakmak and Engin Ergül. Interactions of personal and occupational risk factors on hand grip strength of winter pruners. *International Journal of Industrial Ergonomics*, 67(June):192–200, 2018. 1

[6] Junjie Cao, Andrea Tagliasacchiy, Matt Olsony, Hao Zhangy, and Zhixun Su. Point cloud skeletons via Laplacian-based contraction. In *2010 Shape Modeling International Conference*, pages 187–197, 2010. 3

[7] Erick Castellanos, Felix Ramos, and Marco Ramos. Semantic death in plant's simulation using lindenmayer systems. In *10th International Conference on Natural Computation, ICNC 2014*, pages 360–365. IEEE, 2014. 2

[8] Margarita N Favorskaya and Lakhmi C Jain. Realistic Tree Modelling. In *Handbook on and Geographic Remote Sensing Advances in Information Systems*, volume 122, chapter 6, pages 397–415. Springer, 2017. 2, 3

[9] Lixian Fu, Ji Liu, Jianling Zhou, Min Zhang, and Yan Lin. Tree Skeletonization for Raw Point Cloud Exploiting Cylindrical Shape Prior. *IEEE Access*, 8:27327–27341, 2020. 2, 3

[10] Thomas Guillon, Yves Dumont, and Thierry Fourcaud. A new mathematical framework for modelling the biomechanics of growing trees with rod theory. *Mathematical and Computer Modelling*, 55(9-10):2061–2077, 2012. 2

[11] Hui Huang, Shihao Wu, Daniel Cohen-or, Minglun Cong, Hao Zhang, Guiqing Li, and Baoquan Chen. L1-Medial Skeleton of Point Cloud. *ACM Transactions on Graphics*, 32(4):1–8, 2013. 2

[12] Ken James and N Haritos. Branches and damping on trees in winds. In *23rd Australasian Conference on the Mechanics of Structures and Materials*, pages 1–6, 2014. 2

[13] Wei Jiang, Kai Xu, Zhi Quan Cheng, and Hao Zhang. Skeleton-based intrinsic symmetry detection on point clouds. *Graphical Models*, 75(4):177–188, 2013. 4

[14] Anthony King. The future of agriculture. *Nature*, 544(7651):S21–S23, 2017. 1

[15] Chang Liu, Dezhao Luo, Yifei Zhang, Wei Ke, Fang Wan, and Qixiang Ye. Parametric Skeleton Generation via Gaussian Mixture Models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[16] Yotam Livny, Zhanglin Cheng, Feilong Yan, Baoquan Chen, Soeren Pirk, Oliver Deussen, and Daniel Cohen-Or. Texture-Lobes for Tree Modelling. *ACM Transactions on Graphics*, 30(4):1–10, 2011. 2

[17] Yotam Livny, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Transactions on Graphics*, 29(6):1, 2010. 2

[18] Javier Lluch, Emilio Camahort, Jose Luis Hidalgo, and Roberto Vivo. A hybrid mutiresolution representation for fast tree modeling and rendering. In *International Conference on Computational Science, ICCS 2010*, pages 485–494. Elsevier, 2010. 2

[19] Valeriano Méndez, Heliodoro Catalán, Joan R. Rosell, Jaume Arnó, Ricardo Sanz, and Ana Tarquis. SIMLIDAR - Simulation of LIDAR performance in artificially simulated orchards. *Biosystems Engineering*, 111(1):72–82, 2012. 2

[20] Małgorzata Młotek, Łukasz Kuta, Roman Stopa, and Piotr Komarnicki. The Effect of Manual Harvesting of Fruit on the Health of Workers and the Quality of the Obtained Produce. *Procedia Manufacturing*, 3(Ahfe):1712–1719, 2015. 1

[21] Hiromi Mochiyama. Model validation of discretized spatial closed elastica. *IEEE International Conference on Intelligent Robots and Systems*, 2016-Novem:5216–5223, 2016. 6

[22] Erdal Ozbay, Ahmet Cinar, and Zafer Guler. A hybrid method for skeleton extraction on Kinect sensor data: Combination of L1-Median and Laplacian shrinking algorithms. *Measurement: Journal of the International Measurement Confederation*, 125:535–544, 2018. 3

[23] Oleg Panichev, Ciklum Ukraine, and Alona Voloshyna. U-Net based convolutional neural network for skeleton extraction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[24] Przemyslaw Prusinkiewicz and Pierre Barbier De Reuille. Constraints of space in plant development. *Journal of Experimental Botany*, 61(8):2117–2129, 2010. 2

[25] Przemyslaw Prusinkiewicz, Mikolaj Cieslak, Pascal Ferraro, and Jim Hanan. Modeling Plant Development with L-Systems. In Richard Morris, editor, *Mathematical Modelling in Plant Biology*, chapter 8, pages 139–169. Springer, 2018. 2

[26] Przemyslaw Prusinkiewicz and James Hanan. *Lindenmayer systems, fractals, and plants*. Springer Science & Business Media, 2013. 2

[27] Ed Quigley, Yue Yu, Jingwei Huang, Winnie Lin, and Ronald Fedkiw. Real-Time Interactive Tree Animation. *IEEE Transactions on Visualization and Computer Graphics*, 24(5):1717–1727, 2018. 2, 3, 5

[28] Rafsan Ratul, Shaheena Sultana, Jarin Tasnim, and Arifinur Rahman. Applicability of space colonization algorithm for real time tree generation. *2019 22nd International Conference on Computer and Information Technology, ICCIT 2019*, 1:1–6, 2019. 3

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 3

[30] Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz. Modeling trees with a space colonization algorithm. In *Eurographics Workshop on Natural Phenomena*, pages 63–70, 2007. 2, 3

[31] Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, 2016. 4

[32] Tatsumi Sakaguchi and Jun Ohya. Modeling and animation of botanical trees for interactive virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 139–146, 1999. 2

[33] Ilya Shlyakhter, Max Rozenoer, Julie Dorsey, and Seth Teller. Reconstructing 3D tree models from instrumented photographs. *IEEE Computer Graphics and Applications*, 21(3):53–61, 2001. 2

[34] Ruben M. Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. A survey on procedural modelling for virtual worlds. *Computer Graphics Forum*, 33(6):31–50, 2014. 1

[35] Hanns Christof Spatz and Benoit Theckes. Oscillation damping in trees. *Plant Science*, 207:66–71, 2013. 2

[36] O. Stava, S. Pirk, J. Kratt, B. Chen, R. Měch, O. Deussen, and B. Benes. Inverse procedural modelling of trees. *Computer Graphics Forum*, 33(6):118–131, 2014. 2

[37] Abdel Aziz Taha and Allan Hanbury. An Efficient Algorithm for Calculating the Exact Hausdorff Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2153–2163, 2015. 6

[38] J. Wither, F. Boudon, M. P. Cani, and C. Godin. Structure from silhouettes: A new paradigm for fast sketch-based design of trees. *Computer Graphics Forum*, 28(2):541–550, 2009. 2

[39] Yili Zhao. *Plant Substructuring and Real-time Simulation Using Model Reduction*. PhD thesis, University of Southern California, 2014. 2