

# Creative Virtual Tree Modeling through Hierarchical Topology-preserving Blending

Yutong Wang, Xiaowei Xue, Xiaogang Jin, *Member, IEEE*, Zhigang Deng, *Senior Member, IEEE*

**Abstract**—We present a new method to efficiently generate a set of morphologically diverse and inspiring virtual trees through hierarchical topology-preserving blending, aiming to facilitate designers' creativity production. By maintaining the topological consistency of the tree branches, sequences of similar yet different trees and novel intermediate trees with encouragingly interesting structures are generated by performing inner-species and cross-species blending, respectively. Hierarchical fuzzy correspondences are automatically established between two or multiple trees based on the multi-scale topology tree representations. Fundamental blending tasks including *morph*, *grow* and *wilt* are introduced and organized into a tree-structured blending scheduler, which not only introduces the randomness into the blending procedure but also wisely schedules the tasks to generate topology-aware blending sequences, contributing to a variety of resulting trees that exhibit diversities in both geometry and topology. Most significantly, multiple batches of blending can be executed in parallel, resulting in a rapid creation of a large repository of diverse trees.

**Index Terms**—Creative modeling, Tree modeling, Hierarchical topology preservation, and Shape blending.

## 1 INTRODUCTION

TREES are ubiquitous in both reality and virtual worlds. Despite that tree modeling has been studied for decades, to the best of our knowledge, efficiently generating a series of trees exhibiting inspiring attributes has been largely under-studied. Taking roadside trees in real life as an example, in spite of their apparent similarities at the first glance, they are indeed diverse in both geometry and topology. Therefore, designers have to spend non-trivial efforts to create such different trees while preserving certain high-level similarities. Additionally, in films (e.g., *The Corpse Bride*, and the series of *Harry Potter*) and games (e.g., *Trine*, *Ori and the Blind Forest*), trees demonstrating bizarre geometries or structures usually boost up the ethereal atmosphere in virtual environments. However, modeling a single uncanny tree structure from scratch calls for inspiration and wild imagination. Although a growth-based approach [1] was recently proposed to interactively alter the appearance of an existing tree, it was not designed to be "creative", i.e. able to automatically create interesting trees. Therefore, it generally takes designer's tedious efforts to obtain a desired output tree, let alone creating a large number of different trees. Consequently, a number of previous approaches have been proposed to facilitate the designer's creativity, including [2], [3], [4], [5]. Nevertheless, none of them targets tree modeling or needs to tackle the technical challenge of preserving a strictly hierarchical topology continuity.

Inspired by the work of [5], we resort to blending techniques to efficiently generate a pool of novel and intriguing trees to stimulate the creativity of designers. Widely used in computer animation, *shape blending* transforms one shape



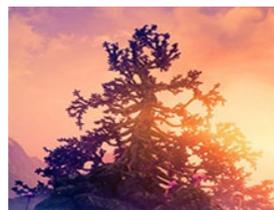
(a) The Whomping Willow



(b) The Ori and the Blind Forest



(c) The Corpse Bride



(d) The Trine

Figure 1: Uncanny trees in popular films and games. Figure 1(a) is from *Harry Potter and the Chamber of Secrets*, Figure 1(b) is a screen shot from *Ori and the Blind Forest*, Figure 1(c) is from *The Corpse Bride*, and Figure 1(d) is from *The Trine*.

into another via maintaining fluid in-between transitions. Although it is not always a necessity to preserve the functional integrity of intermediate shapes, the nature of blending suggests its potential of creating a sequence of new and meaningful intermediate shapes when controlled wisely. In this work, we leverage blending techniques to generate a wide range of inspiring and diverse trees.

We need to tackle two technical challenges for tree blending. The foremost challenge is to find an appropriate representation of correspondences between trees. Previous approaches on measuring the similarity between tree structures [6], [7] generally establish accurate correspondences

- Yutong Wang, Xiaowei Xue, and Xiaogang Jin are with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China  
E-mail: catherineytw@yahoo.com, raxowei@163.com, jin@cad.zju.edu.cn
- Zhigang Deng is with the Department of Computer Science, University of Houston, Houston, TX, USA 77204.



Figure 2: A sequence of blended trees. The top row shows a series of morphologically similar yet different trees (middle) obtained by blending between two *Salix* species (left and right), while the bottom row demonstrates our capability of generating inspiring novel trees by cross-species blending between an artistic *Bonsai* (left) and a *Salix* (right).

between branches by minimizing their structural and geometric differences. However, with a focus on generating novel shapes through blending techniques, we opt to match branches by preserving a certain degree of geometric and topological differences between them. Therefore, *fuzzy* correspondences need to be introduced to maintain a certain degree of variations between matched branches in order to achieve the diversity of the blended shapes. The second challenge is how to keep a record of the topological changes while preserving the hierarchical consistency of the tree branches during the blending process. To handle this, a new blending algorithm needs to be developed.

In this paper, we present a skeleton-based tree blending approach by focusing on generating novel and inspiring blended tree shapes while maintaining their topological continuity and botanical plausibility. Both similar yet different trees and inspiring novel trees are generated through inner-species blending and cross-species blending, respectively (see Figure 2). Besides, our approach also has the ability to perform the  $\mathcal{N}$ -way blending for multiple input trees, see Figure 9.

Figure 3 shows the schematic pipeline of our approach using an example of blending between two trees. At the first step, we abstract the input trees into skeleton-radii representations; see Figure 3(a). To facilitate the preservation of topological hierarchies, we mark the skeletons with the *Gravelius* order [8] followed by building multi-scale topological trees (in a graph-theoretical sense); see Figure 3(b). Then, hierarchical *fuzzy* correspondences are further established between branches; see Figure 3(c), where corresponding ones are rendered in the same color while unmatched chains are rendered in grey. Finally, blending tasks induced by the correspondences are stochastically executed in a topology-aware way, leading to the diversity of blended trees between batches. To facilitate designers’ production, inspiring ones are automatically selected and presented to the designers as artistic inspirations, e.g., the circled trees in Figure 3(d).

**Contributions.** The main contribution of our work is a novel paradigm for virtual tree modeling through hierarchical topology-preserving blending. Although blending techniques have been applied to create novel man-made shapes [5], neither did they explicitly handle the preservation of hierarchical topologies during blending process, nor support interactive freeform exploration of the blending s-

pace spanned by more than two examples. Our work reports the first creative tree modeling method that fulfills the automation of generating a variety of trees from limited input to facilitate designers’ creativities. Not only is our approach capable of generating similar yet different in-between trees, but also it demonstrates the ability to facilitate designers to produce a variety of inspiring novel trees differing both in geometry and topology. Both inner-species and cross-species blending are supported. A novel hierarchical *fuzzy* correspondence establishment scheme is specifically developed for trees, along with a topology-aware blending algorithm that not only maintains the topological continuity among branches but also controls how the sequence of blending tasks is executed to produce trees with diversities. To take one step further, we also propose a method for  $\mathcal{N}$ -way blending, which takes multiple topologically-varying trees (of the same or different species) as the input and effectively spans a “blending” space where designers are allowed to freely explore all the novel trees interpolated by the input in real time.

## 2 RELATED WORK

**Shape blending.** Extensive literature exists in the field of shape blending. Based on different shape representations, existing techniques can be roughly divided into the following three categories: surface based methods [9], [10], [11], scalar fields based methods [12], [13], and implicit functions based methods [14], [15]. Even though the majority of the surface-based methods cope with the blending problem under the assumption of well-aligned shapes with equivalent global topologies, several methods (e.g., [16], [17], [18]) are able to tackle the blending of shapes with different topologies by interpolating either their proxy surfaces or skeletons. Both scalar field and implicit function based methods are capable of dealing with the topological differences of input shapes. Unfortunately, when converting both input and intermediate shapes to certain algorithm-friendly structures, they are often computationally inefficient. In a nutshell, most of existing blending methods only take blending as an appearance-altering tool, while they seldom incorporate high level information (e.g., semantics and functionalities of shapes) into the blending process, often resulting in either discontinuous topologies or less functionally meaningful intermediate shapes.

**Tree modeling.** Despite many progresses on tree modeling in the past several decades, it remains an active research area. Generally, tree modeling techniques fall into two categories: real world tree recovery [19], [20] and virtual tree construction. The latter is more relevant to our approach. Concentrating on simulating the morphology of trees, early  $L$ -systems are capable of modeling a variety of complex trees by parallelly rewriting components from given initial states. However, they suffer from the non-trivial tuning of a large set of parameters and the lacking of direct control over final results. To overcome these, later researchers proposed various interactive modeling methods to give designers more direct control over the generation procedures [1], [21], [22], [23], [24]. Besides, procedural methods [7], [25] inversely estimate developmental parameters from input trees and then stochastically synthesize new trees similar to the input. As a response to the call on intuitive and efficient tree modeling tools, sketch-based tree modeling methods [26], [27], [28], [29] have also been developed in the past.

Various tree representations have also been explored. For example, researchers developed a skeleton-lobe representation to minimize the storage of trees while preserving their global appearances [20]. The work of [30] prefers a skeleton-radius representation by taking advantage of the tree’s cylindrical structure. Similarly, the work of [31] builds bounding volume hierarchies for trees by considering that any tree branches can be bounded by line swept spheres.

Animation systems have also been designed to animate and capture the growth of trees. The X-frog system [22] interpolates pre-defined keyframes, describing the developmental stages of the tree, to generate animated sequences. Pirk et al. [1] proposed a reverse growth method by first estimating the developmental stages of a given model and then interpolating the growth space to create animations. Although these methods tend to animate the morphogenesis of trees, none of them is capable of capturing the transitional morphology of trees when transforming one into another. The work of [31] is relevant to our approach; it introduced a method to generate a multi-family of trees by combining given template trees in a component-wise manner. However, none of the above methods aims to generate inspiring novel trees from examples, or to tackle the problem of combining trees with different topologies.

**Creative modeling.** Since Kreavoy and colleagues’ pioneering work [32], numerous data-driven methods have been proposed to generate novel shapes from examples. Assembly-based approaches rely on high level shape features (e.g., semantics, symmetry, etc.) and create novel shapes by either part-replacement or composition [3], [33], [34]. Also, both probabilistic-based methods [4], [35] and grammar-based methods [36] have been developed to determine the compatibility of different components accurately.

To support the creativity of designers in a greater degree, several recent works present generative systems to create novel shapes via blending techniques. The system in [37] emphasizes designer control over the modeling process, and produces a variety of distinct shapes using a feature-based blending technique. The work of [5] concentrates on a structure-aware automatic blending of topologies and functionalities (e.g., global/partial symmetry, topology continuity, etc), and produces inspiring man-made shapes that

are diverse both in geometry and topology. Instead of using a pre-segmented database, our method achieves the on-the-fly tree segmentation and automatically establishes hierarchical correspondences between branches. In addition, based on the hierarchical topology of trees, our method exploits a multi-scale topological tree to keep a record of topological changes during the blending process, which is more sophisticated than tracking the connections between parts in a structural graph. The topological consistency is maintained all the way to the end of the blending procedure by a tree-structured blending scheduler. We even take one step further by achieving cross-species blending of multiple topologically-varying trees. In sum, our method is advantageous in blending shapes with hierarchical topologies, and is equally capable of generating inspiring new shapes at fine granularity, even superior in achieving  $\mathcal{N}$ -way blending.

### 3 APPROACH OVERVIEW

Blending naturally generates a series of novel intermediate shapes. In this work, we treat blending as a technique of shape generation rather than animation for visual effects. Our method takes two or multiple trees as input, then strategically interpolates them to produce a sequence of in-between trees with topologically continuous branches. Figure 3 shows the key steps of our method.

**Skeletonization.** Taking advantage of the cylindrical nature, input trees are first abstracted into a representation of skeletons and radii, which is more tractable and effective to track their topological changes than the traditional surface representations; see Figure 3(a).

**Segmentation.** Based on the *Gravelius* order [8], we categorize consecutive branches sharing the same order into chains, which segments a tree into botanically meaningful components; see Figure 3(b).

**Fuzzy correspondences.** After constructing a multi-scale topology tree to preserve the hierarchical relationships among chains, we formulate the correspondence establishment as a hierarchical minimum weight bipartite matching problem, where either *one-to-one* or *one-to-none* fuzzy correspondence is set up for a pair of chains; see Figure 3(b).

**Blending.** A hierarchical topology-aware blending algorithm, which stochastically selects and executes blending tasks without violating their topological continuities, is developed to generate novel in-betweens; see Figure 3(c). Due to the randomness, tasks are executed in a stochastic order between batches, resulting in a wild variety of trees. Most importantly, our method is highly parallel by nature, implying that multiple batches of blending can be executed at the same time, contributing to a rapid generation of a large pool of geometrically and topologically varying trees. In Figure 3(d), three batches of blending are performed simultaneously, generating diverse intermediate trees.

**Inspiring tree sampling.** After expurgating seriously colliding branches from the resulting trees, they are further sampled based on their inspiring scores. To the end, a series of novel and interesting trees are selected and presented to designers as inspirations, e.g., the circled trees in Figure 3(d).

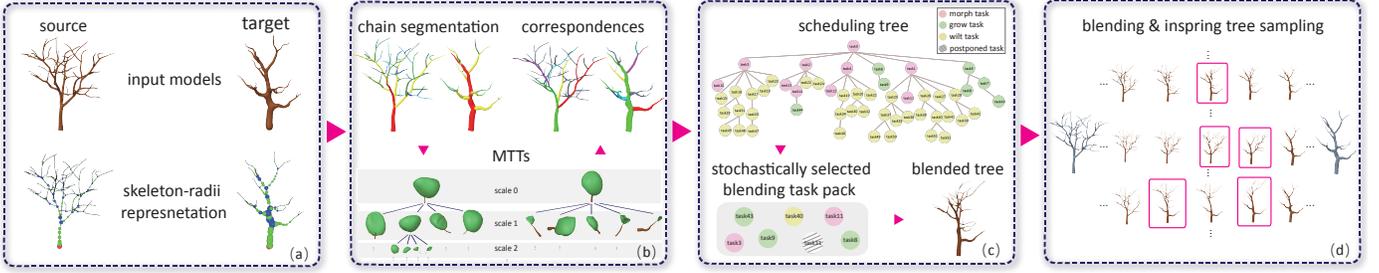


Figure 3: An overview of our method. Skeleton-radii representations of input trees (a), chain segmentation, multi-scale topology trees construction and *fuzzy* correspondence establishment (b), stochastic blending task scheduling (c), and multiple batches of blending and the selection of inspiring trees (circled in red) (d).

## 4 ALGORITHMS

### 4.1 Multi-scale Topological Graph

**Skeleton-radii representation.** Given input trees (see Figure 4(a)), line skeletons (illustrated in Figure 4(b)) are first extracted by the Laplacian contraction method proposed in [38], where a mapping between skeletal nodes and surface is additionally established. Particularly, the corresponding surface forms a spherical shape around each skeletal node. By averaging the distances between a skeletal node and its surrounding surface vertices, we associate each skeletal node with the radius of the spherical shape.

**Gravelius order.** In botanical literature, the *gravelius* order has been widely used to mark branching structures for decades [39]. Therefore, in this work we adopt the branch marking concept by assigning every branch a unique *gravelius* order. Starting from the root branch, whose order is assigned with 1 by default, we evaluate the branching angles of its children and assign orders to them as follows: assign the order of its parent to the child with the minimal angle, and assign an incremented order (+1) to the other children. The procedure repeats until every branch in the tree is assigned. Figure 4(c) shows a marked tree.

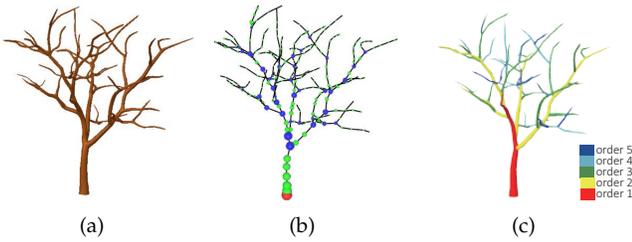


Figure 4: An acer japonicum model (a), the representation of skeleton and radii (b), and a marked model with orders illustrated in color map (c).

**Multi-scale topological tree (MTT).** Following the work of [1], we define a sequence of consecutive branches with the same order as a *chain*. On the basis of the *gravelius* order, chains segment a tree in a botanically meaningful way. Similar to Boudon’s decomposition graph [23], we formalize the input tree into a chain-level multi-scale topological tree (in graph-theoretical sense). At the coarsest scale, we abstract a tree’s topology by its dominant chain (i.e. the lowest ordered chain) and crown as shown in Figure 5(a). When zooming

into the canopies, we delve into the tree’s topology at a finer scale by observing the details of the crown’s sub-structures. Again, these sub-structures are abstracted by their dominant chains and canopies; see Figure 5(b). This zoom-and-observe process repeats until the finest scale is reached, where all the observed topologies are in the granularity of chains; see Figure 5(c). We organize these scale-dependent chain-crown descriptions into a multi-scale topological tree (MTT). Specifically, we define a node  $n_i$  in MTT as a quintuple  $n_i := \langle chn_i, \mathcal{M}_i, \mathcal{CR}_i, v_i, l_i \rangle$ , where  $chn_i$  is a piecewise linear representation of the chain that branches out in the direction of  $v_i$  at location  $l_i$  relative to its parent (if has any). Instead of using the absolute coordinates, the fact that the geometries of chains constantly change over time during the blending procedure, makes it more tractable to describe  $l_i$  using a dimensionless parametric coordinate, which can be computed by the arc-length parameterization of  $chn_i$ ’s parent.  $\mathcal{M}_i$  defines the chain’s crown shape, which is the silhouette of the chain along with all its substructures. In botany literature, a large variety of approaches have been proposed to describe the shape of a crown, such as using cones, ellipsoids, or convex polyhedra. In honor of the input models’ authenticity, we adopt the convex polyhedra as the representation of the crown. The crown ratio  $\mathcal{CR}_i$  is defined as a ratio of the chain’s crown height to the chain’s length, providing a dimensionless description of the lowest location where the chain might bear its substructure. Rather than indicating feature inheritances from the nodes at the coarser scales as in [23], the edges in the MTT only preserve the hierarchy between chains. Specifically, if the chain denoted by node  $n_j$  is a child of the chain denoted by node  $n_i$ , the MTT connects them using an edge from  $n_i$  to  $n_j$ . Figure 5 illustrates an example of the multi-scaled chain-crown descriptions of a tree and its corresponding MTT.

### 4.2 Hierarchical Fuzzy Correspondences

**Overview.** To the best of our knowledge, none of previous tree modeling works has addressed how to establish *fuzzy* correspondences among hierarchical tree branches as well as how to *conduct a structure-aware blending between them*. To tackle these problems, we formulate the task of finding correspondences as a *minimal weight bipartite matching* problem, and develop a top-down matching strategy based on the hierarchical MTTs.

Given two trees  $\mathcal{S}$  and  $\mathcal{T}$ , apparently, nodes in their MTTs, i.e.  $MTT_{\mathcal{S}}$  and  $MTT_{\mathcal{T}}$ , are mutually disconnected.

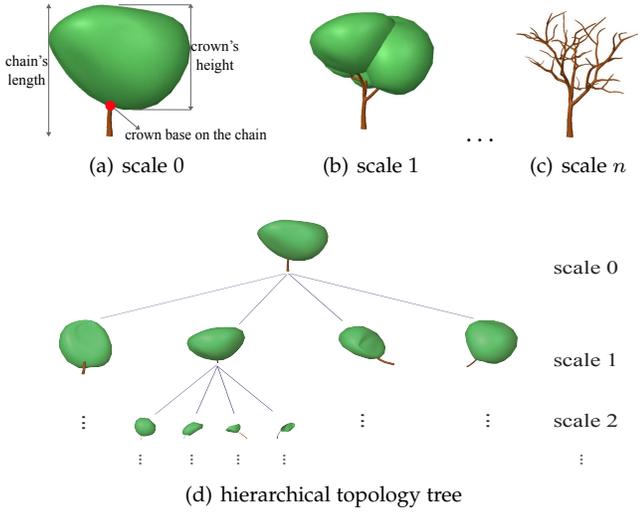


Figure 5: An example of multi-scale topological tree. Scales of descriptions with dominant trunks and their crowns (a)-(c), the corresponding *MTT*(d). For visually pleasing illustrations, the convex polyhedra of canopies are rendered in “spherical” shapes.

Therefore, it is suitable to formulate the establishment of correspondences between  $\mathcal{S}$  and  $\mathcal{T}$  as a minimum weight bipartite matching problem by allowing only *one-to-one* and *one-to-none* matchings between nodes. The problem is solved by minimizing the following objective function in Eq. 1 with the node matching cost function  $\mathcal{C}(\cdot)$ . In this work, we use the well-known *Kuhn-Munkras* algorithm [40], [41] to obtain the optimal solution  $\mathcal{E}(\cdot)$ .

$$\min \sum_{n_i \in MTT_S} \sum_{n_j \in MTT_T} \mathcal{C}(n_i, n_j) \cdot \mathcal{E}(n_i, n_j) \quad (1)$$

$$\mathcal{E}(n_i, n_j) = \begin{cases} 0, & \text{if no match between } n_i \text{ and } n_j \\ 1, & \text{if } n_i \text{ is matched to } n_j \end{cases},$$

subject to

$$\begin{cases} \sum_{n_i \in MTT_S} \mathcal{E}(n_i, n_j) = 1, & \text{if } nbr(MTT_S) \leq nbr(MTT_T) \\ \sum_{n_j \in MTT_T} \mathcal{E}(n_i, n_j) = 1, & \text{if } nbr(MTT_S) > nbr(MTT_T) \end{cases},$$

where  $nbr(\cdot)$  denotes the number of nodes in an *MTT*.

**Matching cost function.** Nodes exhibiting similar attributes are more likely to be matched with less effort than those with large distinctions. Therefore, we defined the cost of matching  $n_i \in MTT_S$  to  $n_j \in MTT_T$  to be inversely proportional to their similarities, which is formally expressed as  $\mathcal{C}(n_i, n_j) = 1 - Sim(n_i, n_j)$ , where  $Sim(\cdot)$  denoted the similarity measurement and is normalized into  $[0, 1]$ . The  $Sim(n_i, n_j)$  is computed by Eq. 2, where  $w_M$ ,  $w_{CR}$ , and  $w_v$  are the weights.  $Sim(\mathcal{M}_{n_i}, \mathcal{M}_{n_j})$  is measured by the *Hausdorff* distance between the crowns, which are pre-aligned according to their lowest locations along the associated chains, i.e. the crown bases on the chains (see Figure 5(a)). We use the *Euclidean* distance to measure the similarities between branching vectors, the  $Sim(v_i, v_j)$ , and

the *Manhattan* distance to measure the similarities between crown ratios, the  $Sim(CR_i, CR_j)$ .

$$Sim(n_i, n_j) = Sim(\mathcal{M}_i, \mathcal{M}_j) \cdot w_M + Sim(v_i, v_j) \cdot w_v + Sim(CR_i, CR_j) \cdot w_{CR} \quad (2)$$

Note that in Eq. 2 we intentionally leave out the measuring of chains’ geometric similarities and their growth locations similarities. It can be explained by our purpose of preserving a certain degree of differences between matched nodes in order to generate novel and interesting chains during the blending procedure. In contrast to the accurate branch similarity measurements [6], [7], in our work, chains with high similarity scores may vary in both geometries and substructures. Therefore, they are *fuzzily* matched.

**Matching algorithm.** Based on the hierarchical *MTT*, two rules are introduced to preserve the topological consistency of matched nodes.

**Rule 1:** Only nodes at the same scale are expected to be matched.

**Rule 2:** Nodes at a finer scale can be matched if and only if their parents at the coarser scale are matched.

The Rule 1, which is the foundation of our matching algorithm, prevents matching nodes at different scales. The Rule 2 implies that nodes at finer scales shouldn’t be matched when their parents are unmatched at the coarser scale, showing a downward propagation of the matching and unmatched labels from the coarser scales to the finer scales in the *MTTs*. Figure 6 illustrates the top-down propagation, where matched nodes are rendered in the same color and unmatched nodes are rendered in grey. Given two matched nodes  $s_2$  and  $t_2$ , on the basis of the matching rules, nodes  $s_3, s_4, s_5, t_3$  and  $t_4$  are the matching candidates at scale 2. After matching, the unmatched node  $s_5$  propagates the unmatched label to its child  $s_7$  at scale 3, which propagates the unmatched label to the child  $t_8$  at scale 4. Since  $s_4$  has no matching candidates at scale 3,  $t_6$ , the child of  $s_4$ ’s corresponding node in *MTT* $_T$ , is unmatched and propagates the unmatched labels to its children  $t_7$  and  $t_8$  at scale 4.

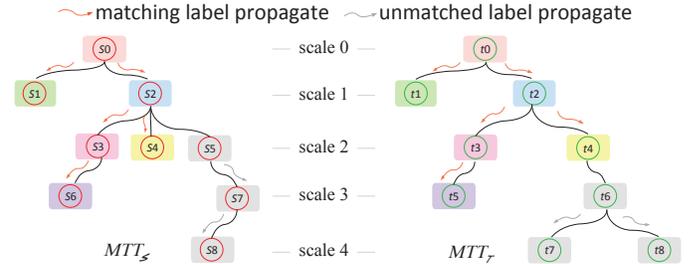


Figure 6: The matching and unmatched labels propagate downwards the *MTTs* in the direction suggested by the orange and the grey arrows, respectively.

Without loss of generality, we define the cost of matching nodes that violate any of the rules as a positive infinity. The matching algorithm works in a top-down fashion, and employs the matching rules to ensure the establishment of topologically consistent correspondences, see Algorithm 1. Starting from the root nodes at the coarsest scale (line 1), the matching between them is trivial. Then, following the

Rule 2, matching and unmatched labels are propagated downwards the  $MTTs$  (line 3). The children of unmatched nodes are assigned with unmatched labels which are further propagated downwards the  $MTTs$ . For the matched nodes, a bipartite graph is constructed with their children at the finer scale (Rule 1 and Rule 2). Again, the algorithm finds the optimal solution to the minimum weight bipartite matching problem, and repeats the propagation-then-matching procedure until it reaches the finest scale of either of the  $MTTs$ . Post-processing is required when  $MTT_S$  and  $MTT_T$  are of different resolutions, and we label all the unassigned nodes with unmatched labels (line 5).

---

**Algorithm 1** Hierarchical matching algorithm
 

---

**Require:** $MTT_S$  and  $MTT_T$ **Ensure:**

Matched pairs of nodes in the form of  $\mathcal{E} = \{ \mathcal{E}(n_i, n_j) \mid n_i \in MTT_S, n_j \in MTT_T \}$ .

- 1: (*Initialization*) Take the coarsest nodes of the source and the target  $MTTs$ , and construct a bipartite graph  $\mathcal{B}^l$ , where  $l$  denotes the scale.
  - 2: (*Matching*) Find a minimum weight matching for  $\mathcal{B}^l$ .
  - 3: (*Propagation*) Assign unmatched label to the children of the unmatched nodes in  $\mathcal{B}^l$ . For the matched nodes, put them into  $\mathcal{E}$ , and build a bipartite graph  $\mathcal{B}^{l+1}$  with their children at the finer scale.
  - 4: (*Loop*) Repeat 2 and 3 until reaches the finest scale of either of the  $MTTs$ .
  - 5: (*Post-process*) In case that  $MTT_S$  and  $MTT_T$  are of different resolutions, label all the unassigned nodes with the unmatched labels.
  - 6: **return**  $\mathcal{E}$ ;
- 

### 4.3 Blending Algorithm

**Overview.** Although a large number of blending techniques have been developed to produce smooth blending results, the issues of preserving the consistency of hierarchical topology and generating functionally meaningful in-betweens have seldom been explicitly handled. Previous interpolation methods may violate the hierarchy exhibiting in the  $MTTs$ , resulting in disconnected structures. To avoid this, we introduce three fundamental blending tasks, and develop a novel blending algorithm, which not only introduces randomness into the blending procedure to achieve diverse blending results, but also maintains the topological continuity of the resulting trees.

**Blending tasks.** Correspondences induce blending tasks. Three types of blending tasks are defined in this work: *grow*, *wilt*, and *morph*, corresponding to the *one-to-none* and *one-to-one* matchings between chains. In order to introduce randomness into the blending procedure, we associate each blending task with an *active* time  $t_{act}$  and a *death* time  $t_{dth}$ . The death time defines the whole life cycle of a blending task, and is randomly initialized to a positive value. The active time, which identifies an active task ( $t_{act} < t_{dth}$ ), records the degree of blending that has been performed. It is initialized to 0 at the beginning of the blending.

Instead of directly adding or removing chains via the splitting and merging operations in [7], we implement the *grow* and *wilt* tasks as continuous operations by progressively interpolate the geometries of involved chains during the blending procedure. Specifically, the *grow* task adds a chain at the moment when  $t_{act} \neq 0$  and grows its geometry during  $t_{act} < t_{dth}$ . Growing a chain  $chn_j \in \mathcal{T}$  to a degree of  $d (= t_{act}/t_{dth})$  is accomplished in two steps. We first grow  $chn_j$  to  $chn_j^d$  by simply preserving  $\|chn_j^d\| = \|chn_j\| \cdot d$ , where  $chn_j^d$  is the blended chain, and  $\|\cdot\|$  denotes the chain's length. Then, we compute the growth point of  $chn_j^d$  on the blended chain of  $chn_j$ 's parent (suppose that it is available) using the growth location  $l_j$ , and link  $chn_j^d$  to the blended chain at the growth point. The *wilt* task is performed similarly since it is the inverse operation of the *grow* task. The geometry of the chain is wilted during  $t_{act} < t_{dth}$ , and it is removed at the moment when  $t_{act} = t_{dth}$ .

The *morph* task induces geometric blending between two corresponding chains, which is achieved by smoothly interpolating their geometries, the branching vectors and the growth locations. Given a pair of matched chains, after aligning them with a rigid transformation estimated from their branching vectors, we sample them with the same number of vertices. The geometry of a blended chain  $chn^d$  is obtained through linear interpolation of these vertices. Besides, the radii of the vertices are first linearly interpolated and then adjusted based on the allometry rule [42] as in Eq. 3 when reconstructing the resulting geometry. The branching vector of  $chn^d$  is computed through quaternion interpolation of the respective source and target branching vectors. In terms of the growth point, we linearly interpolate the chains' growth locations (the parametric coordinates relative to their parents), and then compute the position where  $chn^d$  is linked to the parent. Finally, the newly created chain is rigidly transformed by the interpolated branching vector and linked to its parent at the growth point.

Although more advanced interpolation methods are also available for the same purpose, we experimentally found that the above simple interpolation is sufficient to yield satisfactory results.

**Topology-preserving task scheduling.** At the beginning of the blending procedure, blending tasks are organized into a *scheduling* tree (in graph-theoretical sense), whose nodes are the blending tasks and edges representing their topological hierarchies are inferred from involved chains; see Figure 3(c). The scheduling tree allows us to keep track of structural changes during the blending process and preserve the topological continuity of the blended trees. To avoid generating disconnected structures during blending, we impose two rules on the execution order of the blending tasks.

**Grow rule:** Grow tasks are impended until the parents of the involved chains are available.

**Wilt rule:** Wilt tasks that completely remove the involved chains are not applied until the children of the involved chains are completely wilted.

Rather than applying all the blending tasks simultaneously as in the traditional 2D/3D blending methods, the amount of blending performed at a time in our algorithm is defined by a set of active blending tasks stochastically selected from the scheduling tree, i.e. the blending task

pack; see Figure 3(c). We benefit from the blending task pack in the way that it introduces randomness into the blending procedure and contributes to the diversities of resulting trees. At any time, our blending algorithm first stochastically selects several active blending tasks from the scheduling tree, and then checks whether their execution will violate the blending rules. Based on hierarchies among the blending tasks in the scheduling tree, a *grow* task shouldn't be performed if its parent is also a *grow* task and has not performed any blending, i.e.  $t_{act} = 0$ . A *wilt* task with  $t_{act} = t_{dth}$  should be postponed if its child tasks are still active, i.e.  $t_{act} < t_{dth}$ . A *morph* task with  $t_{act} \neq 0$  is executed under any circumstances since the involved chains (except the root chains) must have available parent as implied by the matching Rule 2. Tasks that conform to the blending rules are put in a blending task pack, and are executed hierarchically to preserve the topological continuity of the blending results. The blended tree is generated as follows: only chains involved in the blending task pack are interpolated, whereas the others remain unchanged. Before applying a blending task, we increase its  $t_{act}$  with a random amount and truncate  $t_{act}$  to  $t_{dth}$  in the case of  $t_{act} > t_{dth}$ . The blended chain is generated by executing the blending task to the degree of  $d = t_{act}/t_{dth}$ , and then linked to the corresponding parent to make the resulting tree topologically continuous. The algorithm repeats until no available active tasks existing in the scheduling tree. Till now, one batch of blending is completed. As a result, a sequence of in-between tree skeletons is generated. Due to the stochastic generation of the blending task pack, blending tasks may be executed in different orders with different amount of degrees between batches, contributing to a variety of blended tree sequences. Most significantly, our blending algorithm is highly parallel, indicating that multiple batches of blending can be performed simultaneously. This increases designers' possibility of discovering inspiring trees by allowing them to explore multiple candidates at the same time.

**Geometry reconstruction.** Since the radii of skeletal nodes are linearly interpolated by the vertices of the corresponding chains, the resulting skeletons might have visually unnatural "bumps" at the branching locations. To overcome this, we employ Murray's allometry rule [42] to adjust the radii of the interpolated skeletons. The relationship between the radius of a parental skeletal vertex and those of its children can be computed by Eq. 3, where  $r_p$  denotes the radius of the parental vertex and  $r_{ci}$  is the radius of the  $i$ -th child. Starting from skeleton terminals, we update their parents' radii using Eq. 3, and recursively apply the allometric rule to their grandparents. The procedure repeats until it finishes adjusting the root radius.

$$r_p^{2.5} = \sum_i^n r_{ci}^{2.5}, \quad (3)$$

The cylindrical nature of tree branches makes it suitable to reconstruct mesh models from the skeleton-radii representations using sweeping techniques. Although more advanced reconstruction methods, including Poisson reconstruction [43] and convolution surface reconstruction [30], are also available for this purpose, we found that as long as the skeletons and radii are correctly computed, even a sim-

ple reconstruction method is sufficient to yield satisfactory results.

**Foliage.** To be visually natural and pleasing, hundreds of thousands of leaves are usually required to represent a foliage. Given a trunk, candidate leaf points are randomly seeded within the volume of its first-scale crown. Served as attraction points, small new twigs are grown on the trunk towards the leaf points via space colonization algorithm [44]. Leaves are randomly oriented and then added along the twigs. We render leaves as textured quads based on the notion of *Billboard Clouds* [45].

#### 4.4 Implausibility Filter

It is noteworthy that the reconstructed trees are not guaranteed to have collision-free branches. In fact, branches might penetrate each other and result in botanically implausible trees. Therefore, an implausibility filter is developed to remove seriously intersecting branches via collision detection. The seriousness of intersection is measured by the ratio of the intersected area to the smaller surface area of two branches. An intersection is considered as serious if the ratio is higher than a user-defined threshold. The threshold is empirically set to 0.01 at the coarsest scale, and linearly increased at finer scales. The reason for using higher thresholds for finer scale branches is that people are less sensitive to colliding branches at the finer scales. Thanks to the cylindrical nature of tree branches, we build a *line swept sphere* bounding volume hierarchy for each tree to speed up the collision detection process.

#### 4.5 Inspiring Tree Sampling

Generally, one batch of blending will generate plenty of intermediate trees. To avoid designers' tedious and repetitive exploration of a large tree gallery, we develop an automatic sampling strategy to select and present most representative and inspiring trees to designers.

We associate each tree an inspiring score, which is defined as the weighted sum of the dissimilarities between the tree and the blending inputs. The dissimilarity between two trees is computed by weighting their geometric and topological differences. Since the global shape of a tree is mainly specified by the silhouette defined by its main trunk and the lateral branches [23], the geometric difference between trees is globally measured by the weighted sum of the dissimilarities between their crowns and the crown ratios at the coarsest scale using the *Hausdorff* distance and the *Manhattan* distance, respectively. Topological difference is measured by the tree-to-tree edit distance [6].

To preserve the diversity of the sampled trees, we perform  $K$ -means clustering on the trees' inspiring scores, and then select a representative tree from each cluster, i.e. the center of cluster, for designers' inspiration. The value of  $K$  is experimentally set as  $\lceil \sqrt{\mathcal{N}/2} \rceil$ , where  $\mathcal{N}$  is the number of trees in the sequence.

#### 4.6 Exploratory $\mathcal{N}$ -way Blending

To help designers explore the greatest possibility of discovering inspiring trees, we develop an exploratory tool which takes  $\mathcal{N} (\geq 2)$  topologically-varying trees as the input and

interactively generates a variety of trees in the blending space spanned by the input. Although the correspondences among the given  $\mathcal{N}$  trees can be obtained by solving a multi-indexed bipartite matching problem, the computational cost would be increasingly prohibitive when  $\mathcal{N}$  becomes large. Instead, we propose a pairwise approximation that generalizes the minimum weight bipartite matching described in Section 4.2 into a  $\mathcal{N}$ -partite problem.

Similar to matching two trees, correspondences among  $\mathcal{N}$  trees (denoted as  $\mathcal{T}_0, \dots, \mathcal{T}_{\mathcal{N}-1}$ ) can also be established in a top-down fashion, as shown in Algorithm 2. First, we group the  $\mathcal{N}$  trees into  $\frac{\mathcal{N} \cdot (\mathcal{N}-1)}{2}$  pairs (line 1), and then calculate the pairwise correspondences by finding the minimal weight bipartite matchings, where *MTTs* are built as by-products to preserve topological hierarchies (line 2). Given the set of pairwise correspondences  $\mathcal{E} = \{\mathcal{E}_{pq} \mid p, q \in [0, \mathcal{N}], p \neq q\}$ , the matching between chains,  $n_i \in \mathcal{T}_p$  and  $n_j \in \mathcal{T}_q$ , is denoted as  $\mathcal{E}_{pq}(n_i, n_j)$ .

Starting from the coarsest scale of all the *MTTs*, a connection graph  $\mathcal{G}^l$ , whose nodes correspond to all the paired chains at the same scale  $l$ , is defined. Node connections are implied from the pairwise correspondences. The inter-pair correspondences are inferred from the transitivity of the pairwise correspondences, e.g.  $\{\mathcal{E}_{pq}(n_i, n_j), \mathcal{E}_{qr}(n_j, n_k)\}$  implies that  $\{\mathcal{E}_{pr}(n_i, n_k)\}$ , corresponding to paths in  $\mathcal{G}^l$ . By the rule of transitivity, the problem of establishing inter-pair correspondences is equivalent to finding all the transitive paths in the connection graph. To maximize the matches among chains, we employ a greedy algorithm by iteratively searching for the longest transitive paths in the graph. Once the inter-pair correspondences are established, we employ the Rule 2 to propagate the matching and unmatched labels downwards the *MTTs* to a finer scale. Topological consistency of the nodes in  $\mathcal{G}^{l+1}$  is ensured by removing node connections if their parents are unmatched. The algorithm repeats until it reaches the finest scale of any *MTTs*. When the *MTTs* are of different resolutions, a post-processing operation is employed to label chains at the unreached scales as unmatched.

With the correspondences, an  $\mathcal{N}$ -way blended tree is hierarchically interpolated as follows. Starting from the coarsest scale, geometry of the blended chain is interpolated by the matched chains with the input weights via the blending tasks (see Section 4.3). For every matched chains at the finer scales, we interpolate their geometries and growth points then link the blended chain to its corresponding parent at the growth point. The procedure repeats until it reaches the finest scale. Note that the number of chains in the blended tree might be higher than any of the inputs due to the greedy establishment of the correspondences. However, it is actually desirable in our work as it blends among multiple chains and increases the chance of generating an inspiring tree by creating more novel branches.

## 5 RESULTS AND EVALUATIONS

We tested our approach on a dataset of 60 trees collected from the Internet (i.e., <http://www.evermotion.org/>), and these trees were constructed by a variety of methods including laser-scanned points, X-frog and *L*-systems.

---

### Algorithm 2 $\mathcal{N}$ -way matching algorithm

---

**Require:**

$\mathcal{N}$  input trees.

**Ensure:**

A set of matched pairs of chains  $\mathcal{E}$ .

- 1: (*Initialization*) Group the  $\mathcal{N}$  input into  $\frac{\mathcal{N} \cdot (\mathcal{N}-1)}{2}$  pairs.
  - 2: (*Pairwise Matching*) For each pair of trees, calculate correspondences using Algorithm 1.
  - 3: (*Inter-pair Matching: step 1*) Starting from the coarsest scale, build a connection graph  $\mathcal{G}^l$  for all the paired chains based on their pairwise correspondences, where  $l$  denotes the scale.
  - 4: (*Inter-pair Matching: step 2*) A greedy algorithm is employed to find inter-pair correspondences by iteratively searching for the longest transitive paths existing in  $\mathcal{G}^l$ .
  - 5: (*Inter-pair Matching: step 3*) Propagate the unmatched labels to the children of the unmatched pairs. Put the matched pairs of chains into  $\mathcal{E}$ , then build a connection graph  $\mathcal{G}^{l+1}$  for their children.
  - 6: (*Loop*) Repeat line 4, 5 until the algorithm reaches the finest scale of any *MTTs*.
  - 7: (*Post-process*) In case that input *MTTs* are of different resolutions, label all the chains at the unreached scales with unmatched labels.
  - 8: **return**  $\mathcal{E}$ ;
- 

The results generated by our approach are shown in Figure 7 to Figure 9, and Figure 15.

**Roadside tree generation.** Blending between trees of the same species creates a series of similar yet different in-between trees; see Figure 7. The resemblance to the input trees make them suitable to be modeled as lines of roadside trees. Additionally, their noticeable differences between each other capture the intrinsic diversity exhibited in real-life trees, even among the same species. Figure 7(b) shows a city scene of roadside trees using the similar yet different trees shown in Figure 7(a).

**Inspiring novel tree creation.** Inspiring novel tree species are generated by blending different kinds of tree species. Figure 8 presents several inspiring trees selected by designers from four galleries of cross-species blending results. The chance of obtaining interesting in-between trees is increased when the input trees differ largely both in geometries and topologies. This can be explained by the establishment of the hierarchical *fuzzy* correspondences, where chains with noticeably different geometries and sub-structures might be matched.

**Exploratory  $\mathcal{N}$ -way blending.** Designers were allowed to perform an  $\mathcal{N}$ -way blending by importing  $\mathcal{N}$  ( $\geq 2$ ) trees into our system, which spontaneously spanned to a blending space. When the designers hand-picked any interesting point in the space (i.e. the red points in Figure 9(a)-(d)), our algorithm interactively generated an in-between tree using the parametric coordinates as interpolation weights of the input. Figure 9 shows several novel trees (brown) blended from four tree species (grey) and their associated blending weights.

One potentially promising application of the resulting inspiring novel trees is to model ethereal scenes where

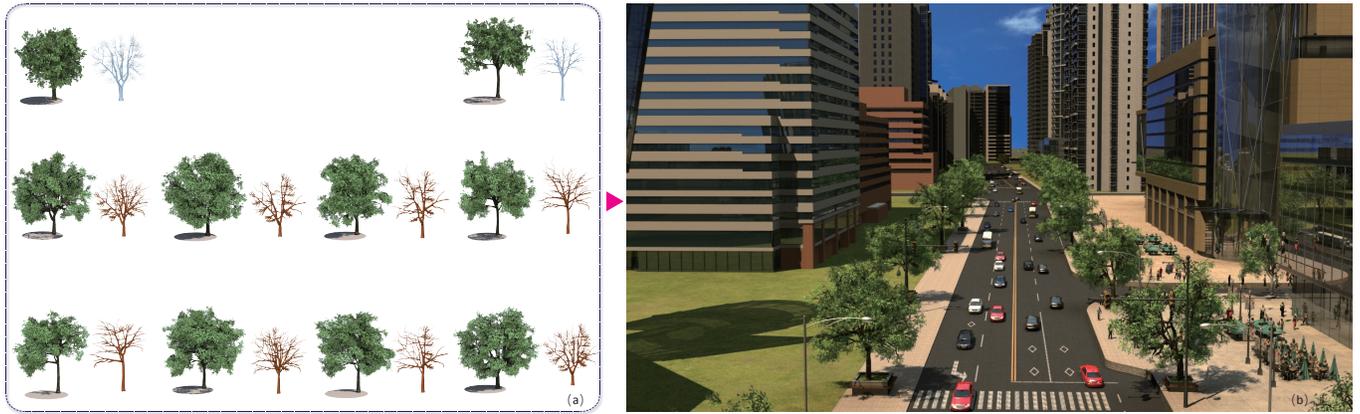


Figure 7: A small gallery of similar yet different new trees (brown) was generated through blending between two *Ulmus* species (grey) (a). (b) is a city scene of roadside trees using trees in (a).



Figure 8: Collections of designer-selected inspiring novel trees (brown) generated by cross-species blending (grey). Note that only two trees are sufficient to generate interesting trees exhibiting distinctive geometric and topological features.

uncanny trees are important to highlight the mysterious atmosphere. Figure 15 shows an eerie ghost house surrounded by bizarre trees generated by the cross-species blending.

**Runtime performance.** We measured the runtime performance of our approach on a desktop equipped with Intel<sup>®</sup> Core i7 clocked at 3.50GHz, 8GB of RAM and

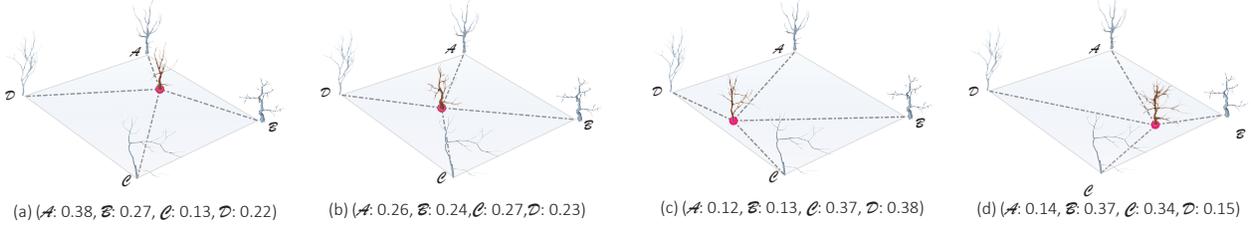


Figure 9: Examples of 4-way blending. Within the blending space spanned by four different input tree species (grey), four inspiring novel trees (brown) are generated by blending with different weights (a)-(d).

NVIDIA<sup>©</sup> Geforce GTX 660 GPU. The pre-processing stages, including skeleton extraction, chain segmentation, and *MTT* construction, took less than five seconds on average. In particular, the majority of the measured time (about 83%) was spent on skeleton extraction. Fuzzy correspondence establishment was completed within a second. It usually took less than twenty seconds to perform one batch of blending, as shown in Table 1. Note that the values of the blended trees in the table are recorded based on the blending results before performing the implausibility filtering and the inspiring tree selection. Most critically, multiple batches can be executed in parallel. In terms of the  $\mathcal{N}$ -way correspondences, it is well known that finding the accurate optimal solution to a multi-indexed matching problem might be  $\mathcal{NP}$ -hard [46]. Instead, the proposed method approximately find the feasible correspondences via pairwise correspondences. The time complexity is  $\mathcal{O}(n^3 \cdot \mathcal{N}^2)$ , where  $n$  is the number of nodes in the pairwise bipartite matching and  $\mathcal{N}$  is the number of input.

**Comparison with topologically-varying blending.** Since few, if any, previous approaches are targeted on the same research problem (i.e., generating either similar yet different or inspiring bizarre trees from limited input), in this work we compare our method with the most related work [5] which is focused on generating topologically-varying, in-between, man-made shapes through blending techniques. As shown in Figure 10, in this comparison, our method produced better results, in particular, showing the ability of producing a variety of intermediate trees without generating topologically disconnected branches.

To quantitatively evaluate the preservation of topology integrity, we defined the topology continuity ratio (*TCR*) of a tree as  $N_{tc}/N_{total}$ , where  $N_{tc}$  and  $N_{total}$  are the total number of topologically continuous branches and the total number of branches of the tree, respectively. In Figure 11, we plotted the *TCR*s of the trees generated by both methods versus the blending process. Evidently, our method (red plot) maintained the topological integrity of blended trees during the whole procedure, whereas [5] (blue plot) failed to preserve the topological continuity of branches and had lower *TCR* values at the majority time of the blending procedure. The main reason is threefold: (1) In terms of structural preservation, the method in [5] emphasized on maintaining the group symmetries of man-made shapes during the blending process. However, the fact that trees typically exhibit insignificant symmetries but significant hierarchies among branches makes it crucial to maintain the topological consistency of branches when performing tree

blending operations. (2) Rather than establishing context-free correspondences using *Hausdorff* distances, we employ a topology-aware *fuzzy* correspondence finding approach based on the *MTT*, hence avoid matching branches with incompatible parents. (3) In addition, the topological continuity of the resulting trees is also maintained by the blending algorithm, which strategically schedules the execution of the blending tasks to avoid generating disconnected branches. On the contrary, the method in [5] simply sampled and then permuted the blending tasks without handling the potentially topology inconsistencies, leading to a clear explanation to the resulting disconnected branches.

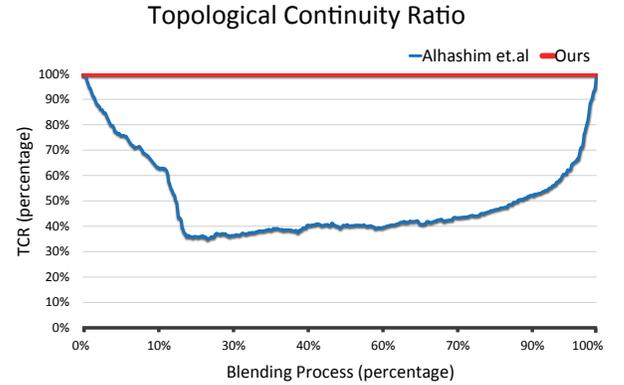


Figure 11: Plots of the *TCR*s of blended trees versus the blending process. Our method (red plot) shows better capability of preserving topology continuity than [5]’s method (blue plot).

## 5.1 User Studies

**Paired comparison with parametric procedural method.** We conducted a paired comparison user study by comparing our method with a state of the art parametric procedural method [7], which suggested the potential of generating similar yet different intermediate trees by manually interpolating the parameters of given input trees. The pairwise comparison was performed on 3 different tree species, each of which had 8 similar yet different trees, including 4 trees generated by our method and the other 4 trees produced by [7]. In this way, there were 4 pairs of comparison for each tree species, and 12 pairs in total. 29 graduated students at a University volunteered to participate in the study. Before the experiment, they were informed that stimuli were

Example	Input chains		Blended Trees (average)	Blending Time (average)
	Source	Target		
<i>Salix</i> (Figure 2 (top))	213	594	317	15.72 sec
<i>Salix-Bonsai</i> (Figure 2 (bottom))	594	32	292	5.04 sec
<i>Ulmus</i> (Figure 7)	348	170	527	12.25 sec
<i>Monterey cypress-Mountain ash</i> (Figure 8 (a))	180	113	152	3.78 sec
<i>Dawn redwood-Artistic Bonsai</i> (Figure 8 (b))	84	32	89	1.32 sec
<i>Maidenhair-Flame maple</i> (Figure 8 (c))	245	307	323	16.71 sec
<i>Whitethorn acacia-Artistic Bonsai</i> (Figure 8 (d))	56	42	127	1.58 sec

TABLE 1: Runtime statistics.



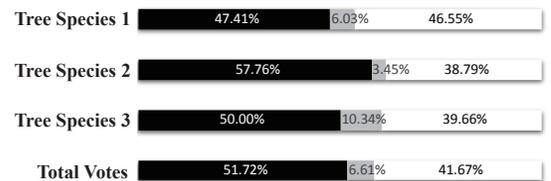
Figure 10: Comparison between our results (bottom) with those obtained by an implementation of [5] (top). The latter produced trees violating the topological continuity (disconnected branches are rendered in dark grey) due to inappropriate correspondences and a lack of decent hierarchy preservation.

structurally similar yet different trees, and they needed to vote for the perceptually natural one between the two trees in a pair. Instead of forcing them to make two-alternative choices, we allowed the participants to choose the *undecided choice* when they couldn't differentiate the two. Based on the previous research study [47], such an experiment setup would help to increase the accuracy and robustness of the study outcomes. To ensure the validity of our study, the trees of the same species were paired randomly and presented to the participants in a random order during the study. The left/right placement of trees within each pair was also randomized to counter balance the subconscious bias of the participants.

We analyzed the study outcomes (see Figure 12) by performing a two-tailed independent one-sample T-test. The corresponding p-value ( $= 0.1943$ ) indicated that there was no statistically significant difference between the results by our method and those by [7]. However, from the raw voting data in Figure 12, we can see that our method clearly received more votes than [7]. This can be explained by the capability of our method to preserve certain substructures of the blended trees exactly the same as the input, as shown in the circled branches in Figure 13. Since the blending is performed stochastically, there is a possibility that several substructures keep their original shapes when either of the following conditions is satisfied: (1) the corresponding blending tasks have never been executed ( $t_{act} = 0$ ); (2) all the corresponding blending tasks have reached their death time ( $t_{act} = t_{dth}$ ). Besides, the stochasticity of the blending algorithm also opens up to a chance of obtaining *partial* blending between inputs. Instead of performing on

the blending task packs that are randomly selected from the scheduling tree, the blending can be only applied to a set of user-defined task packs containing operations on the branches that users are interested in. Consequently, it will produce sequences of *partially* blended trees consisting of both newly generated branches and the unchanged uninterested branches from the inputs. Although [7] has the ability to produce new trees via parameter interpolation, it was hardly possible to reproduce even a small part of the inputs using the probability-based growth model in [7]. In addition, obtaining the parameters of a given input tree took non-trivial effort (over 20 minutes on average). In contrast, our method showed the advantage of rapidly producing a large pool of plausible candidates at an interactive speed.

### Our method VS. Parametric procedural method



■ Scores of our method ■ Scores of *undecided* choices □ Scores of parametric procedural method

Figure 12: The resulting histogram of the paired comparison user study between our method and the method in [7]. The p-value of a two-tailed independent one-sample T-test indicates that there were no statistically significant difference between the results generated by the two methods in terms of the similar yet different quality.

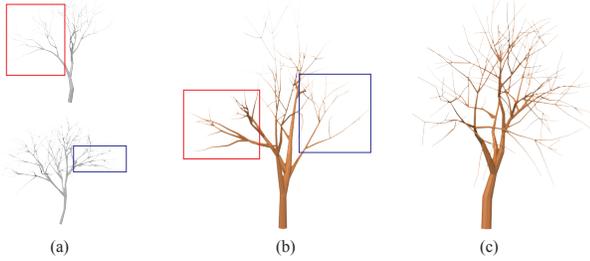


Figure 13: Comparison between the results generated by our *partial* blending (b) and [7] (c), given (a) as the input. Note that the circled structures in (b) maintain the exact form as the source (red) and target (blue) in (a).

**Creativity evaluation.** We performed a second user study to evaluate the *creativity-invoking aspect* of our method in terms of generating inspiring trees. 4 sets of cross-species blended trees were chosen as our stimuli, each of which consists of 10 generated test trees varying in both geometry and topology. We recruited 29 student in a university to participate in the study. Four of them are art students, and the others are graduated students majoring in graphics or digital media. Based on the overall appearance and branching structures, the participants were asked to vote on each generated test tree as *very inspiring*, *inspiring*, *neutral*, *banal*, or *very banal* when compared to the original input. To alleviate the potential issue of visual fatigue, test trees along with the corresponding inputs were randomly presented to the participants, who were given a short break every five tests. We analyzed the voting results, and plotted the percentages of scores in each category, as illustrated in Figure 14. Our method on average received over 50% inspiring scores (including *very inspiring* and *inspiring* scores), whereas less than 15% of discouraging scores (including *banal* and *very banal*). This clearly demonstrated the capability of our method to generate inspiring novel trees from natural trees (the inputs) via blending techniques. It is noteworthy that despite the promising results, our creativity evaluation study is still preliminary; more comprehensive studies need to be conducted to further evaluate the creativity quality, which is a part of our future work.

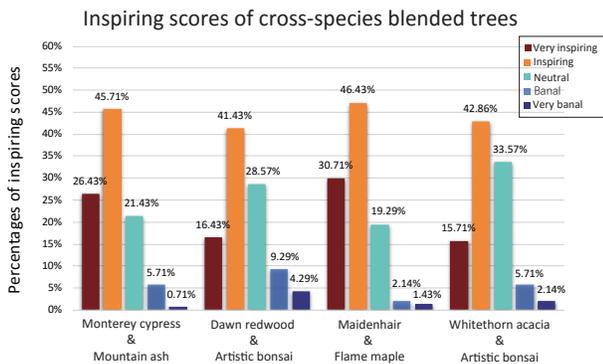


Figure 14: Plots of the inspiring scores of cross-species blended trees generated by our method.

## 6 CONCLUSIONS AND DISCUSSION

In this paper we present the first approach to efficiently generate novel yet inspiring trees with high quality via blending techniques. Our method is highly versatile and can be potentially applied to many unconventional tree modeling scenarios, such as generating similar yet different tree of the same species for real-life roadside tree modeling; see Figure 7, and interestingly bizarre virtual trees for ethereal scenes in movies and games; see Figure 8 and Figure 15. The capability of creating novel in-between trees not only distinguishes our approach from traditional tree modeling methods, but also has a clear advantage over the existing blending-based creative modeling tool [5] (Figure 10) in preserving the topological continuity of the resulting shapes.

**Limitations and Future work.** Although the results by our method are plausible and encouraging, our method still leaves room for improvement. The granularity of topological variations is dependent on the granularity of tree segmentation. In our current work, blending is performed at the chain level, corresponding to the axis scale tree graph in botany literature [48]. Fine-grained geometric and topological variations may be achieved by blending at the growth unit or the inter-node scale. Besides, our current method can only handle macrophanerophytes. Considering the case of blending between a macrophanerophyte (typically having one main stem) and a shrub (typically having multiple main stems), *merging* and *splitting* tasks would be required. Since the goal of our current work is focused on generating as many geometrically and topologically varying trees as possible, problems such as smooth ramification [30] and collision-free tree branches [31] are not explicitly handled, which are a part of our future work. The quality of resulting trees are simply controlled by a conservative implausibility filter. It may yield trees violating certain botanical attributes, such as self-similarity and biomass distribution. Such botanical violations are less acceptable when modeling real-life trees. However, it is greatly desirable if designers attempt to obtain encouraging bizarre trees, where these violations help them to push the envelope of their imagination ever further. Therefore, it is necessary to design a more “intelligent” filter. We leave it as the future work.

To further facilitate creativity, in the future we plan to incorporate more designers’ controls. It would be useful to give designers the flexibility to control the crown shapes and branching patterns during the blending procedure for their artistic needs. One promising solution is to employ sketch-based techniques. Environmental factors, such as sunlight and obstacles, might also affect the blending process, and should be taken into account in the future work. In addition to botanical trees, our method could be potentially applied to other branchy-structures, such as roots and vessels, by incorporating specific concerns raised by the modeling purposes.

## ACKNOWLEDGMENTS

Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant no. 61472351). Zhigang Deng was supported by the Joint Research Fund for Overseas Chinese, Hong Kong and Macao Young Scientists of the



Figure 15: An eerie scene of bizarre trees generated by cross-species blending.

National Natural Science Foundation of China (Grant No. 61328204) and US National Science Foundation (NSF) Grant IIS-1524782.

## REFERENCES

- [1] S. Pirk, T. Niese, O. Deussen, and B. Neubert, "Capturing and animating the morphogenesis of polygonal tree models," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 169, 2012.
- [2] S. Chaudhuri and V. Koltun, "Data-driven suggestions for creativity support in 3d modeling," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 6. ACM, 2010, p. 183.
- [3] K. Xu, H. Zhang, D. Cohen-Or, and B. Chen, "Fit and diverse: set evolution for inspiring 3d shape galleries," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 57, 2012.
- [4] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun, "A probabilistic model for component-based shape synthesis," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 55, 2012.
- [5] I. Alhashim, H. Li, K. Xu, J. Cao, R. Ma, and H. Zhang, "Topology-varying 3d shape creation via structural blending," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 158, 2014.
- [6] K. Zhang, "A constrained edit distance between unordered labeled trees," *Algorithmica*, vol. 15, no. 3, pp. 205–222, 1996.
- [7] O. Stava, S. Pirk, J. Kratt, B. Chen, R. Měch, O. Deussen, and B. Benes, "Inverse procedural modelling of trees," in *Computer Graphics Forum*, vol. 33, no. 6. Wiley Online Library, 2014, pp. 118–131.
- [8] H. Gravelius, *Flusskunde*. GJ göschen, 1914, vol. 1.
- [9] M. Alexa, D. Cohen-Or, and D. Levin, "As-rigid-as-possible shape interpolation," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 157–164.
- [10] T. Michikawa, T. Kanai, M. Fujita, and H. Chiyokura, "Multiresolution interpolation meshes," in *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*. IEEE, 2001, pp. 60–69.
- [11] M. Alexa, "Recent advances in mesh morphing," in *Computer graphics forum*, vol. 21, no. 2. Wiley Online Library, 2002, pp. 173–198.
- [12] D. Cohen-Or, A. Solomovic, and D. Levin, "Three-dimensional distance field metamorphosis," *ACM Transactions on Graphics (TOG)*, vol. 17, no. 2, pp. 116–141, 1998.
- [13] X. Fang, H. Bao, P. A. Heng, T. Wong, and Q. Peng, "Continuous field based free-form surface modeling and morphing," *Computers & Graphics*, vol. 25, no. 2, pp. 235–243, 2001.
- [14] J. F. Hughes, "Scheduled fourier volume morphing," in *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2. ACM, 1992, pp. 43–46.
- [15] G. Turk and J. F. O'brien, "Shape transformation using variational implicit functions," in *ACM SIGGRAPH 2005 Courses*. ACM, 2005, p. 13.
- [16] D. DeCarlo and J. Gallier, "Topological evolution of surfaces," in *Graphics Interface*, vol. 96, 1996, pp. 194–203.
- [17] T. Surazhsky, V. Surazhsky, G. Barequet, and A. Tal, "Blending polygonal shapes with different topologies," *Computers & Graphics*, vol. 25, no. 1, pp. 29–39, 2001.
- [18] R. L. Blanding, G. M. Turkiyyah, D. W. Storti, and M. A. Ganter, "Skeleton-based three-dimensional geometric morphing," *Computational Geometry*, vol. 15, no. 1, pp. 129–148, 2000.
- [19] H. Xu, N. Gossett, and B. Chen, "Knowledge and heuristic-based modeling of laser-scanned trees," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 4, p. 19, 2007.
- [20] Y. Livny, S. Pirk, Z. Cheng, F. Yan, O. Deussen, D. Cohen-Or, and B. Chen, *Texture-lobes for tree modelling*. ACM, 2011, vol. 30, no. 4.
- [21] J. Weber and J. Penn, "Creation and rendering of realistic trees," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 119–128.
- [22] B. Lintermann and O. Deussen, "Interactive modeling of plants," *Computer Graphics and Applications, IEEE*, vol. 19, no. 1, pp. 56–65, 1999.
- [23] F. Boudon, P. Prusinkiewicz, P. Federl, C. Godin, and R. Karwowski, "Interactive design of bonsai tree models," in *Computer Graphics Forum*, vol. 22, no. 3. Wiley Online Library, 2003, pp. 591–599.
- [24] S. Pirk, T. Niese, T. Hädrich, B. Benes, and O. Deussen, "Windy trees: Computing stress response for developmental tree models," *ACM Transactions on Graphics*, vol. 33, no. 6, 2014.
- [25] R. Wang, W. Hua, Z. Dong, Q. Peng, and H. Bao, "Synthesizing trees by plantons," *The Visual Computer*, vol. 22, no. 4, pp. 238–248, 2006.

- [26] T. Ijiri, S. Owada, and T. Igarashi, "The sketch l-system: Global control of tree modeling using free-form strokes," in *Smart Graphics*. Springer, 2006, pp. 138–146.
- [27] X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, and S. B. Kang, *Sketch-based tree modeling using Markov random field*. ACM, 2008, vol. 27, no. 5.
- [28] J. Wither, F. Boudon, M.-P. Cani, and C. Godin, "Structure from silhouettes: a new paradigm for fast sketch-based design of trees," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 541–550.
- [29] S. Longay, A. Runions, F. Boudon, and P. Prusinkiewicz, "Treesketch: interactive procedural modeling of trees on a tablet," in *Proceedings of the international symposium on sketch-based interfaces and modeling*. Eurographics Association, 2012, pp. 107–120.
- [30] X. Zhu, X. Jin, and L. You, "High-quality tree structures modelling using local convolution surface approximation," *The Visual Computer*, vol. 31, no. 1, pp. 69–82, 2015.
- [31] Y.-J. Kim, J.-H. Woo, M.-S. Kim, and G. Elber, "Interactive tree modeling and deformation with collision detection and avoidance," *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 423–432, 2015.
- [32] V. Kreaovoy, D. Julius, and A. Sheffer, "Model composition from interchangeable components," in *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*. IEEE, 2007, pp. 129–138.
- [33] A. Jain, T. Thormählen, T. Ritschel, and H.-P. Seidel, "Exploring shape variations by 3d-model decomposition and part-based recombination," in *Computer Graphics Forum*, vol. 31, no. 2pt3. Wiley Online Library, 2012, pp. 631–640.
- [34] Y. Zheng, D. Cohen-Or, and N. J. Mitra, "Smart variations: Functional substructures for part compatibility," in *Computer Graphics Forum*, vol. 32, no. 2pt2. Wiley Online Library, 2013, pp. 195–204.
- [35] S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun, "Probabilistic reasoning for assembly-based 3d modeling," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 35.
- [36] X. Guo, J. Lin, K. Xu, and X. Jin, "Creature grammar for creative modeling of 3d monsters," *Graphical Models*, vol. 76, no. 5, pp. 376–389, 2014.
- [37] M. Sanchez, O. Fryazinov, T. Vilbrandt, and A. Pasko, "Morphological shape generation through user-controlled group metamorphosis," *Computers & Graphics*, vol. 37, no. 6, pp. 620–627, 2013.
- [38] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 44.
- [39] M. Holton, "Strands, gravity and botanical tree imagery," in *Computer Graphics Forum*, vol. 13, no. 1. Wiley Online Library, 1994, pp. 57–67.
- [40] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [41] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [42] C. D. Murray, "A relationship between circumference and weight in trees and its bearing on branching angles," *The Journal of general physiology*, vol. 10, no. 5, pp. 725–729, 1927.
- [43] M. Kazhdan, M. Bolitho, and Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.
- [44] A. Runions, B. Lane, and P. Prusinkiewicz, "Modeling trees with a space colonization algorithm," *NPH*, vol. 7, pp. 63–70, 2007.
- [45] S. Behrendt, C. Colditz, O. Franzke, J. Kopf, and O. Deussen, "Realistic real-time rendering of landscapes using billboard clouds," in *Computer Graphics Forum*, vol. 24, no. 3. Wiley Online Library, 2005, pp. 507–516.
- [46] F. C. Spieksma, "Multi index assignment problems: complexity, approximation, applications," in *Nonlinear Assignment Problems*. Springer, 2000, pp. 1–12.
- [47] X. Ma and Z. Deng, "Natural eye motion synthesis by modeling gaze-head coupling," in *Virtual Reality Conference, 2009. VR 2009. IEEE*. IEEE, 2009, pp. 143–150.
- [48] C. Godin and Y. Caraglio, "A multiscale model of plant topological structures," *Journal of theoretical biology*, vol. 191, no. 1, pp. 1–46, 1998.



**Yutong Wang** received the B.Sc. degree in software engineering from Chongqing University, P.R. China, in 2012. She is currently a Ph.D candidate at the State Key Laboratory of CAD&CG, Zhejiang University. Her main research interests include creative modeling and sketch-based modeling.



**Xiaowei Xue** is currently an undergraduate student of the College of Computer Science & Technology, Zhejiang University majoring in Digital Media Technology. His research interests include digital geometry processing and special effects simulation.



**Xiaogang Jin** received the B.Sc. degree in computer science and the M.Sc. and Ph.D degrees in applied mathematics from Zhejiang University, P. R. China, in 1989, 1992, and 1995, respectively. He is a professor in the State Key Laboratory of CAD&CG, Zhejiang University. His current research interests include digital geometry processing, geometric modeling, 3D printing, virtual try-on, insect swarm simulation, traffic simulation, implicit surface modeling and applications, creative modeling, sketch-based modeling, and image processing. He received an ACM Recognition of Service Award in 2015. He is a member of the IEEE and ACM.



**Zhigang Deng** is currently a Full Professor of Computer Science at the University of Houston (UH) and the Founding Director of the UH Computer Graphics and Interactive Media (CGIM) Lab. His research interests include computer graphics, computer animation, virtual human modeling and animation, and human computer interaction. He earned his Ph.D. in Computer Science at the Department of Computer Science at the University of Southern California in 2006. Prior that, he also completed B.S. degree in Mathematics from Xiamen University (China), and M.S. in Computer Science from Peking University (China). He is the recipient of a number of awards including ACM ICMI Ten Year Technical Impact Award, UH Teaching Excellence Award, Google Faculty Research Award, UHCS Faculty Academic Excellence Award, and NSFC Overseas and Hong Kong/Macau Young Scholars Collaborative Research Award. Besides the CASA 2014 Conference General Co-chair and SCA 2015 Conference General Co-chair, he currently serves as an Associate Editor of several journals including *Computer Graphics Forum*, and *Computer Animation and Virtual Worlds Journal*. He is a senior member of ACM and a senior member of IEEE.