



SPECIAL ISSUE PAPER

Topologically consistent leafy tree morphing

Yutong Wang¹  | Luyuan Wang¹ | Zhigang Deng² | Xiaogang Jin¹ ¹State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China²Computer Science Department, University of Houston, Houston, TX, USA**Correspondence**

Xiaogang Jin, State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China.

Email: jin@cad.zju.edu.cn

Funding information

National Natural Science Foundation of China, Grant/Award Number: 61472351

Abstract

We present a novel morphing technique to generate pleasing visual effects between 2 topologically varying trees while preserving the topological consistency and botanical meanings of any in-between shapes as natural trees. Specifically, we first efficiently convert leafy trees into botanically inspired chain-lobe representations in an automatic way. With the aid of branching-pattern aware, one-to-many correspondences between branches and leaves, we hierarchically interpolate branches of in-between trees while maintaining their topological consistencies. Finally, we simultaneously interpolate foliage, specifically every single leaf, during the morphing process, avoiding the generation of unpleasant “floating” leaves. We demonstrate the effectiveness of our approach by creating visually compelling tree morphing animations, even between cross-species.

KEYWORDS

shape morphing, special effects, topological consistency, tree modeling

1 | INTRODUCTION

During the past decades, morphing has consistently proven its usefulness in generating breath-taking visual effects and has become an indispensable tool in the special effects and animation industry. Similarly, the task of morphing between topologically varying trees opens up to efficiently creating aesthetically special effects for many unconventional animation applications, such as nurturing the mysterious atmosphere of the extraterrestrial environment in games and movies.

The majority of existing efforts have been put in generating morphing between man-made objects, animals, even human portraits (please refer to Alexa¹ for a comprehensive survey), and they have produced pleasing results. However, none of them is capable of generating the morphological and topological transformations between trees. The reasons are as follows: (a) they do not especially establish topological-aware correspondences between branches; (b) they do not explicitly maintain the hierarchically consistent morphing trajectories, resulting in disconnected “floating” branches during the morphing process; (c) they do not handle the problem of generating smooth transitions between foliage. In addition to the

fluid topological and geometric transformations maintained in the existing works, morphing between tree requires special care on in-between trees’ botanical features, such as cohesive branching patterns, reasonable foliage shapes, and biomass distributions. Indeed, none of the existing methods is able to ensure the preservation of the above high-level tree features during the morphing process.

The main technical challenges to the tree morphing problem are twofold: (a) establishing botanical-aware, topologically consistent correspondences between different tree species; (b) generating botanically plausible foliage morphing trajectories that avoid producing floating leaves during the morphing process.

In this paper, we propose a novel tree morphing technique, which takes two topologically varying leafy trees as input and efficiently generates botanically sound morphing results by preserving the topological consistency of in-between trees; an example result is shown in Figure 1. To the best of our knowledge, our work reports the first approach that seamlessly integrates foliage morphing into the tree animation framework and demonstrates the ability of generating smooth foliage transformations even between different tree

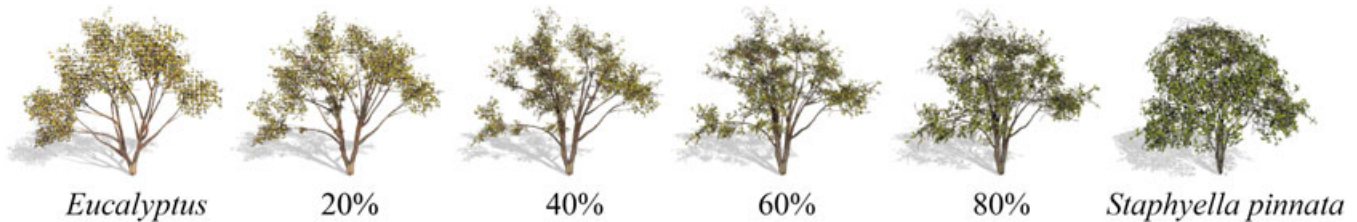


FIGURE 1 Morphing between a *Eucalyptus* and a *Staphylea pinnata* by our approach

species; see Section 7. Our work has three main contributions: (a) an efficient clustering algorithm to automatically segment foliage into botanically plausible groups, that is, *lobes*; (b) a branching pattern-aware correspondence establishment method to allow sound one-to-many matches among branches; (c) a novel foliage morphing algorithm to not only avoid producing unpleasant floating leaves but also produce time-varying foliage details, including geometric and textural transformations of every single leaf.

2 | RELATED WORK

Morphing. To date, a large number of morphing techniques (both 2D and 3D) have been proposed to continuously transform one shape into another. Two-dimensional morphing techniques, including polygonal morphing^{2,3} and image morphing,⁴ suffer from the lack of the three-dimensional information of shapes and may produce unpleasant ghost artifacts when applied to 3D shapes. Therefore, many morphing methods specialized for 3D shapes have been proposed (please refer to Alexa¹ for a comprehensive survey). However, none of the existing methods has explicitly handled the tree morphing problem. The main reason is that morphing between trees requires the preservation of high-level botanical information (such as branching patterns, topological consistency, and botanical plausibility) of the in-between trees, which has not been addressed in all the previous morphing methods.

Tree animations. Despite its long history, the study of tree animations remains to be active. Existing literature can be roughly classified into two categories: tree growth animations and interactions between trees and the environment. The early

X-frog system⁵ allows users to define several keyframes that describe the developmental stages of trees, and then interpolate them to generate tree growth animations. Later, Pirk et al⁶ automatically infer the developmental parameters from one single static tree and then interpolates them to generate the tree growth animations. Environmental factors, including obstacles,^{7,8} lighting,⁸ winds,^{9,10} and competition with other trees¹¹ have been studied to date to realistically animate interactions between trees and the environment. However, none of them is capable of generating fluid morphing between topologically varying trees. The reason is three-fold: the developmental parameters of the growth model are species-dependent; the topologically consistent correspondences between different tree species are hardly addressed; and the smooth transformations between foliage are not considered during the interpolation.

Creative tree modeling. Several exploratory approaches, such as previous studies,^{12–15} have been proposed to assist users to efficiently create high-quality 3D shapes. The recent approach by Wang et al.¹⁶ is the most relevant work to ours. It utilizes blending techniques as a novel shape creation tool, targeting on generating as many morphologically diverse trees as possible. However, it only establishes either one-to-one or one-to-none correspondences between branches because of its bipartite matching algorithm. Without considering the branching patterns, visually unpleasant branches with abnormally large substructures violating the biomass distributions might be generated during the blending process, see Figure 2. Furthermore, they do not tackle the problem of producing a fluid transformation between foliage.

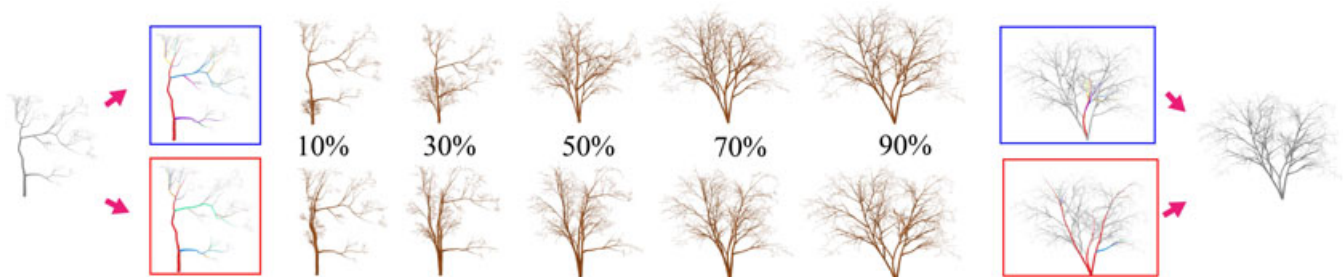


FIGURE 2 Morphing results generated by Wang et al.¹⁶ (top row) and our method (bottom row)

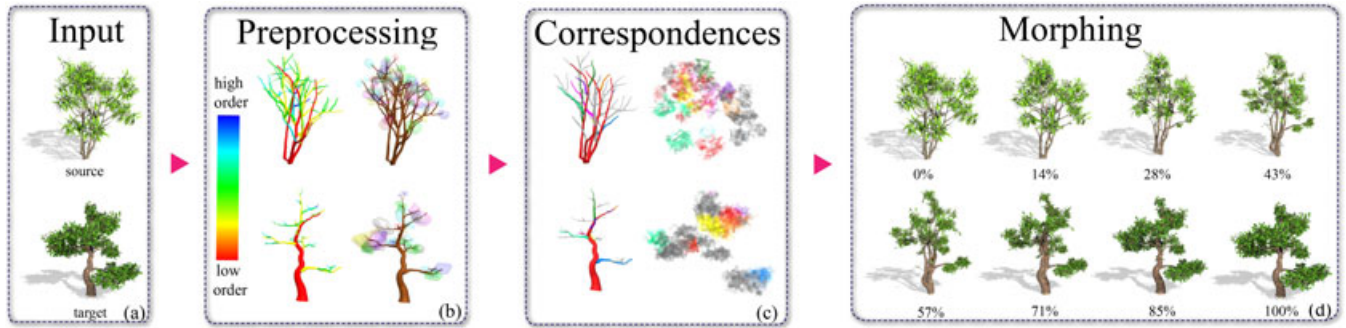


FIGURE 3 Approach overview

3 | APPROACH OVERVIEW

The goal of tree morphing is to generate a sequence of in-between trees that not only fluidly transform from the source to the target but also maintain their topological consistency and botanical plausibility. Figure 3 illustrates the main steps of our approach. Given two leafy trees (Figure 3a), line skeletons are extracted using Oscar et al.,¹⁷ which are further segmented into branching-pattern-aware chain groups based on a hybrid ordering strategy. After constructing the branching-pattern aware topology tree (the BPTT) to preserve the branching patterns and hierarchical topology of chain groups, foliage are automatically clustered into lobes, associating leaves with chains (Figure 3b). Based on the BPTTs, branching-pattern aware, one-to-many correspondences are hierarchically established in two steps: set up either one-to-one or one-to-none correspondences between chain groups by solving a minimum weight bipartite matching problem, and then build one-to-many matches among chains within the matched chain groups using a greedy algorithm. Meanwhile, one-to-many correspondences between leaves within the matched chains are also established by the greedy algorithm (Figure 3c). The in-between leafy trees are hierarchically interpolated by maintaining their topological consistency, resulting in visually pleasing morphing results; see Figure 3(d).

4 | PROCESSING TREE MODELS

4.1 | BPTT

Hybrid branch orderings. Line skeletons of the input trees are first extracted using the Laplacian contraction method.¹⁷ On the basis of the classic strand model,¹⁸ we describe their branching hierarchies using a hybrid of the *gravelius* and the *weibull* orderings. It labels branches sharing the same branching patterns with the same orders, contributing to an efficient identification of branching patterns.

Branch orders are hierarchically assigned from the root branches (with the order 1) towards tip branches. Given a

branch p with ordering $D(p)$, the branching angles and radii of its child branches $\{c_i\}$ are computed, and the ordering $D(c_i)$ is assigned using Equation 1:

$$D(c_i) = \begin{cases} D_w(c_i), & \text{if } \sigma_\alpha \leq \theta \text{ and } \sigma_r \leq r_t, \\ D_g(c_i), & \text{otherwise,} \end{cases} \quad (1)$$

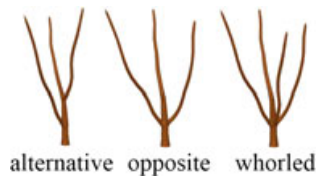
where σ_α and σ_r denote the standard variance of the branching angles and radii, respectively. θ and r_t are the user-defined thresholds, and $D_g(\cdot)$ and $D_w(\cdot)$ are the *gravelius* and *weibull* ordering functions; see Equation 2.

$$D_g(c_i) = \begin{cases} D(p), & \text{if } \alpha_{c_i} = \alpha_{\min}, \\ D(p) + 1, & \text{if } \alpha_{c_i} \neq \alpha_{\min}, \end{cases} \quad (2)$$

$$D_w(c_i) = D(p),$$

where α_{c_i} is the branching angle and α_{\min} is the minimum branching angle. The algorithm repeats until it reaches tip branches.

BPTT. In real world, the growth of branches generally exhibits one of three patterns¹⁹ (see inset). With alternative branching, exact one branch grows at a branching point. In contrast, with opposite and whorled branching, multiple branches grow at the same branching point and are arranged into special forms. Therefore, they should be considered as a group rather than individuals.



We use BPTT, to describe the topological hierarchies among branches. Similar to Pirk et al.⁶ and Wang et al.,¹⁶ consecutive branches sharing the same ordering are defined as *chains*, which are then described by seven parameters listed in Table 1. Branching patterns are identified using Equation 3, which cluster chains into groups, denoted as a set of GC_i :

$$\begin{cases} \text{alternative,} & \text{if } \# = 1, \\ \text{opposite,} & \text{if } \# = 2 \text{ and } |\overline{\alpha_{dvg}} - 180^\circ| < \theta, \\ \text{whorled,} & \text{otherwise} \end{cases} \quad (3)$$

where $\#$ denotes the number of chains sharing the same ordering while growing at the same location, $\overline{\alpha_{dvg}}$ is the average of divergence angles α_{dvg} between chains (Figure 4a), and θ is a user-defined threshold, which is experimentally set to 30° in this paper.

TABLE 1 Parameters of chains

Name	Description
chn	piecewise linear representation of its geometry, including radii information
OL	the shape of the outer lobe
OR	outer lobe ratio, denoting the lowest location where the substructures grow
IL	the shape of the inner lobe
IR	inner lobe ratio, denoting the location where the leaves grow
v	branching vector
p	growth location

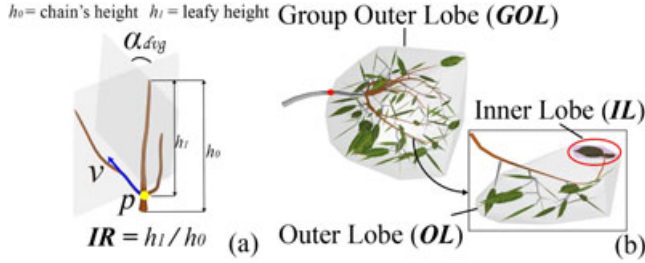


FIGURE 4 Lobes (a) and chain parameters (b)

In the spirit of the multiscale topology tree,¹⁶ whose nodes represent chains and edges encode their hierarchies, we take one step further to preserve the hierarchy of branching patterns and define the nodes of the BPTT as a quintuple:

$$N_i := \langle GC_i, GOL_i, GR_i, Gv_i, Gp_i \rangle,$$

where GC_i is the group of chains exhibiting a certain branching pattern, GOL_i denotes the group outer lobe of the group, GR_i is group lobe ratio, Gv_i is the average branching vector of the chains, and Gp_i is the growth location of the group. N_i and N_j are connected by a BPTT edge if all the chains encoded by N_j could find their parents in N_i .

4.2 | Lobe geometry

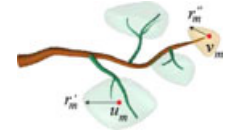
We observe that people subconsciously tend to perceive trees' foliage by dividing it into smaller parts according to their branches, namely, the lobes. The existing lobe extraction method²⁰ usually takes nontrivial efforts to obtain an ideal set of lobes because the thresholds differentiating the foliage from the branches are needed to be manually set and tuned according to the tree species. Instead, we propose an automatic outer-to-inner lobe extraction method, which hierarchically groups leaves into lobes using K -means clustering.

Because leaves grow on the branches, we assume that the foliage of a chain generally follows a crownlike shape formed by its substructures. Three types of lobes are defined in our approach; see Figure 4(b). The outer lobe of a chain (OL) is defined as the leaves covering all of its substructure. The group outer lobe (GOL), describing the general appearance of a chain group's foliage, equals to the union of OL s of the

chains within the group. The inner lobe (or simply the lobe), denoted as IL , defines the leaves growing at the tip of the chain, that is, the actual leaves belonging to a chain.

Our goal is to extract IL s for chains. In order to maintain the botanical consistency, IL s are extracted hierarchically downwards the BPTT in the following three steps.

Step 1: initialization. Starting from the root node N_i of the BPTT, the algorithm initializes its GOL^l as all the leaves in the foliage, where l denotes the level of the node in the BPTT, and $l = 0$ for the root node. Then, the algorithm collects its children and computes the center u_m and radius r'_m of the crownlike shape formed by the substructures of chains for the m th child chain group, which are the candidate centers and radii of their potential foliage. Besides, for any chain $chn_n \in GC_i$, the algorithm assigns its potential foliage center v_n as the tip of the chain, and the radius r''_n as 10% of the chain's length (refer to inset).

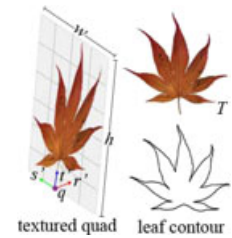


Step 2: clustering. Leaves within the GOL^l are iteratively divided into $(m + n)$ groups by K -means clustering method, where m is the number of child chain groups and n is the number of chains contained in GC_i . Initially, cluster centers are set to be $U \cup V$, where $U = \{u_0, u_1, \dots, u_m\}$ and $V = \{v_0, v_1, \dots, v_n\}$. Unlike the standard K -means clustering algorithm, we associate each cluster with a distance threshold to reduce the influence of outliers, which is experimentally initialized as 35% of the radius r'_i/r''_i and increased by 10.5% during every iteration. For each unassigned leaf, the algorithm measures its Euclidean distances to all the clusters and assigns it to the nearest cluster when the distance is smaller than the distance threshold. Then, the algorithm updates the cluster centers and relaxes the distance threshold by 10.5%. The clustering repeats until all of the leaves within the GOL^l are assigned.

Step 3: iteration. As a result, leaves assigned to clusters that are initialized with $u_i \in U$ define the global outer lobe GOL_j^{l+1} of the child group and serve as the input of the next iteration, whereas the others are the inner lobes of the chains contained in the group GC_i . Then, the algorithm moves downwards the BPTT and repeats the procedure for the child groups at level $l + 1$. The algorithm terminates when all of the leaves are grouped into IL s.

4.3 | Leaf geometry

The foliage usually consists of thousands of leaves. Due to the large amount, we adopt the notion of billboard clouds²¹ and generalize leaves into textured quads. Specifically, a leaf is represented as a septuple as $l_i := \langle chnid_i, q_i, w_i, h_i, r'_i, s'_i, t'_i, T_i \rangle$, where $chnid_i$ represents the chain it grows on, q_i is the leaf position relative to its corresponding chain, w_i and h_i are the



width and height of the leaf, respectively, r'_i, s'_i, t'_i together denotes the leaf's 3D orientation, and T_i denotes its texture.

5 | CORRESPONDENCES

Considering the variety of branching patterns in trees, we establish one-to-many (including one-to-one matching) and one-to-none matchings between chains, see Figure 3(c), where corresponding chains are rendered in the same color. Other than constraining the matching candidates to be nodes at the same level (Rule 1) and have matched parents (Rule 2),¹⁶ we also require that only those chains belonging to the matched groups are supposed to be matched in order to maintain their topological consistency (Rule 3).

The cost of matching two BPTT nodes is measured by their similarities (see Equation (4)):

$$C(N_i, N_j) = \sum_k w_k \cdot \|n_i^k - n_j^k\|, \quad (4)$$

where $\|\cdot\|$ denotes a distance function, n_i^k and n_j^k are the k th element in node N_i and N_j , respectively, and w_k is the weight term. Specifically, we use the Hausdorff distance, the classic shape similarity measurement, to evaluate the similarities between two *GOLs*. Similarities between branching vectors, *GRs*, number of chains within the groups, and average divergence angles are computed by the Euclidean distances. Without loss of generality, matching nodes that violate any of the rules is defined to be a positive infinity. Likewise, the similarity between chains is defined by the weighted sum of similarities between their feature parameters in Table 1.

Similar to Wang et al.,¹⁶ we employ a top-down matching strategy, which hierarchically establishes correspondences based on the BPTTs. However, the correspondences at each level are established via a two-phase protocol in order to preserve their branching patterns: (1) constrained by the Rules 1 and 2, establish either one-to-one or one-to-none correspondences between the BPTT nodes (i.e., the chain groups) by solving a minimum weight bipartite matching problem as Wang et al.¹⁶; (2) according to the Rule 3, greedily match chains with the most similar ones for the matched groups, resulting in one-to-many correspondences.

Correspondences between chains also indicate matches between their *ILs*. Leaves within the matched *ILs* are greedily matched by finding the most similar ones. The similarity between two leaves is equivalent to the weighted sum of similarities between their leaf positions, widths, heights, and orientations.

6 | MORPHING

6.1 | Morphing trunks

Correspondences induce morphological transformations. Five transformation operations are defined in this work; see Figure 5. Inspired by Wang et al.,¹⁶ we apply morph operations to the one-to-one matched chains and grow or wilt operations to chains, which have no corresponding sources or targets. The transformation of one-to-many matched chains is achieved by split or merge operations. The in-between trunks are interpolated and then reconstructed using the method of Alswais et al.¹¹

Specifically, a *p split* of a chain is performed in four steps: (a) make p copies of the chain, denoted as $SP = \{chn_0, chn_1, \dots, chn_p\}$; (b) at time $t \in [0, 1]$, segment $\forall chn_i \in SP$ at the arc-length parametric location $(1-t)$ into two parts, the $chn_i^{(1-t)}$ and the chn_i^t ; (c) compute the interpolation weights for every points n'_j on each part as

$$w'_j = \begin{cases} b \cdot t, & \text{if } n'_j \in chn_i^{(1-t)}, \\ t, & \text{if } n'_j \in chn_i^t, \end{cases}$$

where $b \in (0, 1]$ is the slow-down coefficient, which is proportional to the arc-length parametric value of the point n'_j ; (d) perform morph operation to each copy using the point weight w'_j s. Being the inverse operation of the split task, the merge of chains is performed similarly except that we progressively speed up the morphing of the nodes nearby the chains' roots by calculating the interpolation weight w_j using $\min(1.0, (2.0 - b) \cdot t)$ when $n'_j \in chn_i^{(1-t)}$.

6.2 | Morphing foliage

The in-between leaves are computed by interpolating each term of their feature vectors. The leaf position is equivalent to the linearly interpolated parametric location on the newly morphed chain. The width and height of the leaf are also linearly interpolated, and the orientations are computed through quaternion interpolation. In order to avoid ghosting artifacts, we separately interpolate contours and textons of the corresponding leaves, and then superimpose the morphed texton on the morphed contour and create the in-between leaf. Specifically, leaf contours are extracted by the Canny boundary

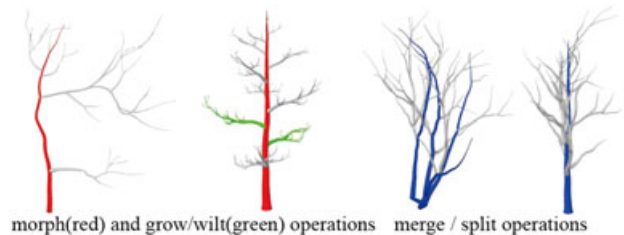


FIGURE 5 Transformation operations

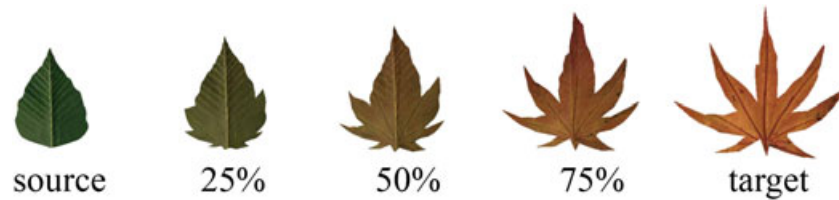


FIGURE 6 Morphing between two leaves

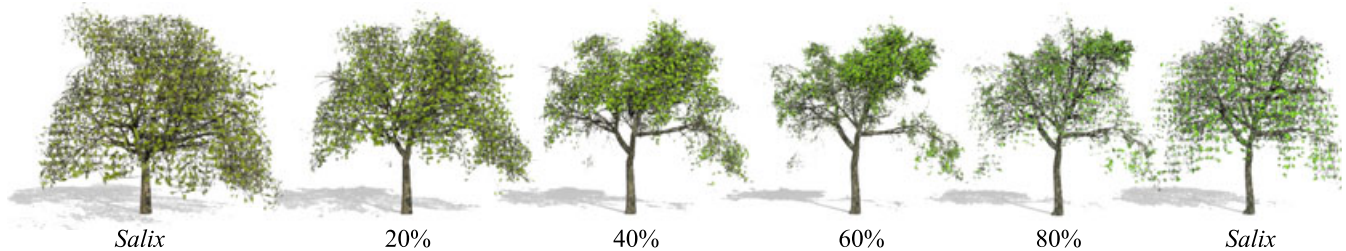


FIGURE 7 Morphing between two *Salix*



FIGURE 8 Morphing between a Whitethorn acacia and an *Acer japonicum*

detection method.²² With the piecewise linear polygon representations, the in-between leaf contours are interpolated using 2D polygon morphing method.³ The in-between textures are simply computed by pixel-wise linear interpolation. Although more advanced leaf modeling method²³ could also be used for our purpose, we experimentally found that our simple interpolation is sufficient to generate visually pleasing results; see Figure 6.

7 | RESULTS AND EVALUATIONS

We collected 60 real-world trees, including arbors* and shrubs†, from the internet (i.e., <http://www.evermotion.org/>). In addition, 10 designers' hand-modeled virtual trees are also incorporated into the dataset to evaluate the effectiveness and flexibility of our approach. With randomly paired inputs, our algorithm is capable of generating visually pleasing and botanically plausible animations for both inner species (Figure 7) and cross-species morphings (Figure 1 and Figure 8). Specifically, branching patterns are maintained

during the morphing process, contributing to natural transitions between matched branches; see Figure 1 and Figure 2. Furthermore, the interpolation of leaves, including shapes and textures, is performed in parallel to the trunk morphing process based on the leaf–branch mappings, generating smoothly transformed foliage sequences. Although the morphing of flowers, fruits, and other factors are not explicitly tackled currently, they can be interpolated in a similar way as that of leaves, see Figure 9, where the source leafy tree perched with butterflies smoothly transforms into the target tree blooming purple flowers. One potential application of the cross-species tree morphing is to enhance the supernatural atmosphere when modeling the extraterrestrial environment in games and movies. Please refer to our animation results in the supplementary video.

Runtime statistics. We have implemented our morphing algorithm in C++ on a desktop equipped with Intel® Core i7 clocked at 3.50 GHz, 8 GB of RAM, and NVIDIA® Geforce GTX 660 GPU. The runtime statistics are presented in Table 2. In sum, our method demonstrates its efficiency of generating smooth morphing results on an off-the-shelf computer.

Comparison with topological-aware tree-blending method. Figure 2 compares our approach with the most relevant work by Wang et al.,¹⁶ which generates diverse inspiring novel trees by blending among topologically varying trees.

*An arbor is a woody plant that have an elongated, dominate stem, that is, the trunk.

†A shrub is a small woody plant that usually has multiple stems forming a decurrent architecture.

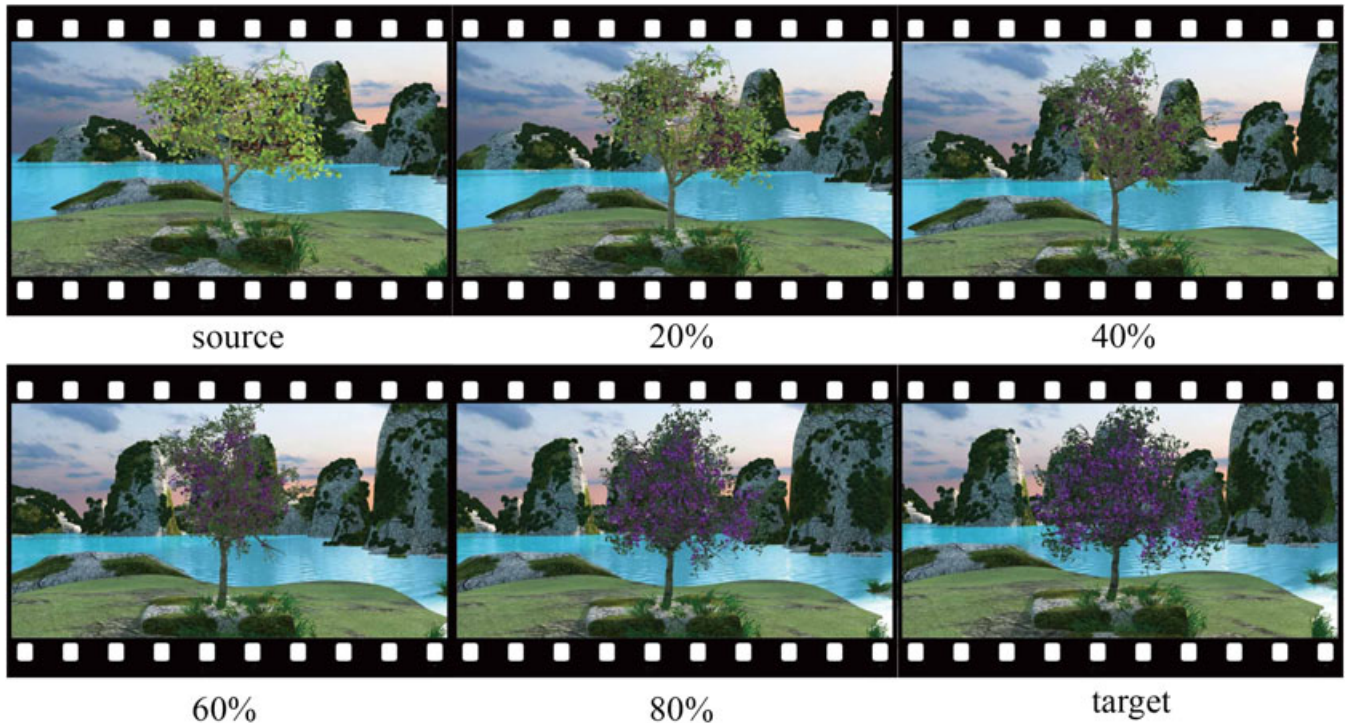


FIGURE 9 A morphing scene, where the source leafy tree perched with butterflies smoothly transforms into the target tree blooming purple flowers

TABLE 2 Runtime statistics

Examples	Time						
	Input statistics		Preprocessing		Set up matches (s)	Morphing (per frame, s)	
	# of chains	# of leaves	Ordering (s)	Lobe extraction (s)			
Figure 1	5,220/20,195	4,690/5,919	1.82	3.12	0.49	11.35	
Figure 3	4,507/4,590	4,436/3,765	0.61	0.99	0.19	1.52	
Figure 8	5,660/4,370	2,937/2,844	1.34	0.97	0.55	7.31	
Figure 7	7,782/21,138	6,459/2,874	4.39	1.81	0.85	7.92	
Figure 9	5,224/8,272	2,351/5,564	1.64	1.61	0.81	10.08	

In contrast to massively growing branches from the main trunk using the method in Wang et al.¹⁶ (top row), our algorithm gradually splits the source main trunk into multiple stems, producing a more natural morphing sequences (bottom row).

In addition to visual comparison, we quantify the “naturalness” of the morphing sequences using the trunk support weight in the well-known pipe model theory.²⁴ Denoted as $T(z)$, it defines the weight sustained by a trunk as all the weights of leaves and branches from the top of its substructures to a given location z along the trunk. Because people are sensitive to the dominant trunks of trees, we compute the $T(OR)$ s for the main trunks of every in-between tree and take the heaviest weight as the maximum main trunk support weight (*MTSW*). Without loss of generality, we normalize the *MTSW* by the total weight of the tree. Figure 10 plots the *MTSW*s of the in-between trees generated by both methods against the morphing process. Evidently, our method (red

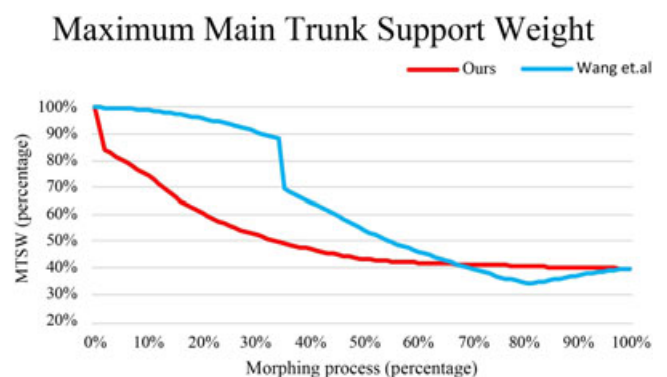


FIGURE 10 Comparison of the *MTSW*s of the morphing sequences generated by our method and¹¹

plot) smoothly “shifts” the weight of leaves and branches sustained by the source main trunk to multiple target stems under the splitting operations. By contrast, the results by Wang et al.¹⁶ (blue plot) shows steep weight drops at the interval of [30%, 40%] and a slow raise at the interval of [70%, 100%].



FIGURE 11 Comparison between our method (top) and the baseline method (bottom)

The above can be explained by our branching pattern preserving one-to-many correspondences (see the red circled trees in Figure 2), where the multiple stems exhibiting the whorled branching patterns are grouped and interpolated at the same time during the whole morphing process, leading to naturally sharing of the trees' weights. In contrast, the correspondence establishment scheme in Wang et al.¹⁶ only allows either one-to-one or one-to-none matches between branches; see the blue circled tree in Figure 2. Therefore, two other unmatched main stems should be interpolated by the grow operations during the morphing process. Because of their young age and small radii, the gradually grown chains are supported by the main trunk during early morphing time (e.g., [0%, 30%]). In the middle time (e.g., [30%, 40%]), when their radii reach a certain threshold, the ordering strategy marks them as main stems. As a result, they share the weight originally supported by the main trunk, leading to the sudden drop of the *MTSW*. Because the tree weight are nonuniformly shifted to growing stems, the *MTSWs* might be lower than the target ones, see the *MTSW* values at the later time (e.g., [70%, 100%]). In a nutshell, our method not only preserves branching patterns during the morphing process but also maintain a smooth *MTSW* change, contributing to more natural and pleasant animations compared to Wang et al.¹⁶

Comparison with baseline foliage morphing method. Because few, if any, existing works have put explicit efforts on the smooth morphing between foliage, we implemented a naive foliage morphing method as the baseline by linearly interpolating randomly matched leaves and compared it with our method, see Figure 11. Apparently, the baseline method produces floating leaves (bottom row, circled leaves), leading to noticeable visual artifacts.

8 | CONCLUSION

In this paper, we present the first approach to efficiently generate smooth morphing sequences between two leafy trees. Unlike the conventional tree modeling techniques,^{6,8,16} which

usually emphasize trunk modeling but treat foliage modeling as postprocessing operations, we seamlessly integrate trunk morphing and foliage morphing into a unified animation framework and generate a novel type of visual special effect. We also develop an automatic foliage segmentation method to efficiently associate leaves with branches using the structure lobe. Our method is advantageous in preserving both geometric and topological consistency of the in-between trees during the morphing process. Therefore, it is highly flexible to create visually compelling morphing effects between topologically varying leafy trees.

Limitations and future work. Our current method still leaves room for improvement. Because our goal is to generate visually pleasing morphing effects, issues such as smooth ramifications²⁵ and collision-free branches²⁶ are not explicitly addressed. In terms of the lobe extraction, environmental factors, such as lighting conditions, wind effects, and so forth, should be considered in the future. In addition, we also plan to incorporate user interaction to our morphing pipeline, where users may control the general crown shapes of the in-between shapes to meet their artistic needs.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their constructive comments that helped improve this paper. Xiaogang Jin was supported by the National Natural Science Foundation of China (Grant 61472351).

REFERENCES

1. Alexa M. Recent advances in mesh morphing. *Comput Graphics Forum*. 2002;21(2):173–198.
2. Carmel E, Cohen-Or D. Warp-guided object-space morphing. *The Visual Comput*. 1998;13(9):465–478.
3. Liu L, Wang G, Zhang B, Guo B, Shum Heung-Yeung. Perceptually based approach for planar shape morphing. *Proc Pacific Graphics*. 2004:111–120.

4. Wolberg G. Image morphing: a survey. *The Visual Comput.* 1998;14(8):360–372.
5. Lintermann B, Deussen O. Interactive modeling of plants. *IEEE Comput Graphics Appl.* 1999;19(1):56–65.
6. Pirk Sören, Niese T, Deussen O, Neubert B. Capturing and animating the morphogenesis of polygonal tree models. *ACM Trans Graphics (TOG).* 2012;31(6):169.
7. Palubicki W, Horel K, Longay S, et al. Self-organizing tree models for image synthesis. *ACM Trans Graphics (TOG).* 2009;28(3):58.
8. Pirk S, Stava O, Kratt J, et al. Plastic trees: interactive self-adapting botanical tree models. *ACM Trans Graphics.* 2012;31(4):50.
9. Zhao Y, Barbič J. Interactive authoring of simulation-ready plants. *ACM Trans Graphics (TOG).* 2013;32(4):84.
10. Pirk S, Niese T, Hädrich T, Benes B, Deussen O. Windy trees: computing stress response for developmental tree models. *ACM Trans Graphics (TOG).* 2014;33(6):204.
11. Alsweis M, Deussen O. Procedural techniques for simulating the growth of plant leaves and adapting venation patterns. *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*; 2015. p. 95–101.
12. Talton JO, Gibson D, Yang L, Hanrahan P, Koltun V. Exploratory modeling with collaborative design spaces. *ACM Trans Graphics (TOG).* 2009;28(5):167.
13. Jain A, Thormählen T, Ritschel T, Seidel H-P. Exploring shape variations by 3d-model decomposition and part-based recombination. *Comput Graphics Forum.* 2012;31(2pt3):631–640.
14. Guo X, Lin J, Xu K, Jin X. Creature grammar for creative modeling of 3d monsters. *Graphical Models.* 2014;76(5):376–389.
15. Guo X, Lin J, Xu K, Chaudhuri S, Jin X. Customcut: On-demand extraction of customized 3d parts with 2d sketches. *Comput Graphics Forum.* 2016;35(5):89–100.
16. Wang Y, Xue X, Jin X, Deng Z. Creative virtual tree modeling through hierarchical topology-preserving blending. *IEEE Trans Visual Comput Graphics.* 2016. <https://doi.org/10.1109/TVCG.2016.2636187>.
17. Au OscarK-C, Tai C-L, Chu H-K, Cohen-Or D, Lee T-Y. Skeleton extraction by mesh contraction. *ACM Trans Graphics (TOG).* 2008;27(3):44.
18. Holton M. Strands, gravity and botanical tree imagery. *Comput Graphics Forum.* 1994;13(1):57–67.
19. Dirr MA. *Manual of woody landscape plants: their identification, ornamental characteristics, culture, propagation and uses*, stipes publ. Co., Champaign, Illinois 554; 1990.
20. Livny Y, Pirk S, Cheng Z, et al. Texture-lobes for tree modelling. *ACM Trans Graphics (Proc SIGGRAPH 2011).* 2011;30(4):53.
21. Behrendt S, Colditz C, Franzke O, Kopf J, Deussen O. Realistic real-time rendering of landscapes using billboard clouds. *Comput Graphics Forum.* 2005;24(3):507–516.
22. Canny J. A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell.* 1986;8(6):679–698.
23. Mundermann L, MacMurchy P, Pivovarov J, Prusinkiewicz P. Modeling lobed leaves. *Proceedings of CGI, 2003.* p. 60–65.
24. Shinozaki K, Yoda K, Hozumi K, Kira T. A quantitative analysis of plant form—the pipe model theory: I. basic analyses. *Jpn J Ecol.* 1964;14(3):97–105.
25. Zhu X, Jin X, You L. High-quality tree structures modelling using local convolution surface approximation. *The Visual Comput.* 2015;31(1):69–82.
26. Kim Y-J, Woo J-H, Kim M-S, Elber G. Interactive tree modeling and deformation with collision detection and avoidance. *Comput Anim Virtual Worlds.* 2015;26(3-4):423–432.



Yutong Wang received her BSc degree in software engineering from Chongqing University, China, in 2012. She is currently a Ph.D candidate at the State Key Lab of CAD&CG, Zhejiang University. Her main research interests include creative modeling, computer animation, and sketch-based modeling.



Luyuan Wang received her BSc degree in digital media from Hunan University, China, in 2016. She is currently a Ph.D candidate at the State Key Lab of CAD&CG, Zhejiang University. Her main research interests include creative modeling and

virtual try-on.



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 to 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 being 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.



Xiaogang Jin is a professor of the State Key Lab of CAD&CG, Zhejiang University, China. He received his BSc degree in computer science in 1989, MSc and PhD degrees in applied mathematics in 1992 and 1995, respectively, all from Zhejiang

University. His current research interests include traffic simulation, insect swarm simulation, physically based animation, cloth animation, special effects simulation, implicit surface computing, computer-generated marbling, nonphotorealistic rendering, and digital

geometry processing. He received an ACM Recognition of Service Award in 2015. He is a member of the IEEE and ACM.

How to cite this article: Wang Y, Wang L, Deng Z, Jin X. Topologically Consistent Leafy Tree Morphing. *Comput Anim Virtual Worlds*. 2017;28:e1762. <https://doi.org/10.1002/cav.1761>