Pole Dancing: 3D Morphs for Tree Drawings

Elena Arseneva[1](✉), Prosenjit Bose[2], Pilar Cano[2,3], Anthony D'Angelo[2], Vida Dujmović[4], Fabrizio Frati[5], Stefan Langerman[6], and Alessandra Tappini[7]

[1] St. Petersburg State University (SPbU), Saint Petersburg, Russia
ea.arseneva@gmail.com
[2] Carleton University, Ottawa, Canada
jit@scs.carleton.ca, anthonydangelo@cmail.carleton.ca
[3] Universitat Politècnica de Catalunya, Barcelona, Spain
m.pilar.cano@upc.edu
[4] University of Ottawa, Ottawa, Canada
vida@cs.mcgill.ca
[5] Roma Tre University, Rome, Italy
frati@dia.uniroma3.it
[6] Université libre de Bruxelles (ULB), Brussels, Belgium
stefan.langerman@ulb.ac.be
[7] Università degli Studi di Perugia, Perugia, Italy
alessandra.tappini@studenti.unipg.it

Abstract. We study the question whether a crossing-free 3D morph between two straight-line drawings of an n-vertex tree can be constructed consisting of a small number of linear morphing steps. We look both at the case in which the two given drawings are two-dimensional and at the one in which they are three-dimensional. In the former setting we prove that a crossing-free 3D morph always exists with $O(\log n)$ steps, while for the latter $\Theta(n)$ steps are always sufficient and sometimes necessary.

We here refer to pole dancing as a fitness and competitive sport. The authors hope that many of our readers try this activity themselves, and will in return introduce many pole dancers to Graph Drawing, thereby alleviating the gender imbalance in both communities. The authors do not condone any pole activity used for sexual exploitation or abuse of women or men.

E. A. was partially supported by F.R.S.-FNRS and SNF grant P2TIP2-168563 under the SNF Early PostDoc Mobility program. P.C. was supported by CONACyT, projects MINECO MTM2015-63791-R and Gen. Cat. 2017SGR1640. P.B, A.D and V.D. were supported by NSERC. F.F. was partially supported by MIUR Project "MODE" under PRIN 20157EFM5C and by H2020-MSCA-RISE project 734922, "CONNECT". S.L. is Directeur de Recherches du F.R.S.-FNRS. A.T. was partially supported by the project "Algoritmi e sistemi di analisi visuale di reti complesse e di grandi dimensioni" - Ric. di Base 2018, Dip. Ingegneria, Univ. Perugia.

T. Biedl and A. Kerren (Eds.): GD 2018, LNCS 11282, pp. 371–384, 2018.
https://doi.org/10.1007/978-3-030-04414-5_27

1 Introduction

A *morph* between two drawings of the same graph is a continuous transformation from one drawing to the other. Thus, any time instant of the morph defines a different drawing of the graph. Ideally, the morph should preserve the properties of the initial and final drawings throughout. As the most notable example, a morph between two planar graph drawings should guarantee that every intermediate drawing is also planar; if this happens, then the morph is called *planar*.

Planar morphs have been studied for decades and find nowadays applications in animation, modeling, and computer graphics; see, e.g., [11,12]. A planar morph between any two topologically-equivalent[1] planar straight-line[2] drawings of the same planar graph always exists; this was proved for maximal planar graphs by Cairns [8] back in 1944, and then for all planar graphs by Thomassen [16] almost forty years later. Note that a planar morph between two planar graph drawings that are not topologically equivalent does not exist.

It has lately been well investigated whether a planar morph between any two topologically-equivalent planar straight-line drawings of the same planar graph always exists such that the vertex trajectories have low complexity. This is usually formalized as follows. Let Γ and Γ' be two topologically-equivalent planar straight-line drawings of the same planar graph G. Then a morph $\mathcal{M}$ is a sequence $\langle \Gamma_1, \Gamma_2, \dots, \Gamma_k \rangle$ of planar straight-line drawings of G such that $\Gamma_1 = \Gamma$, $\Gamma_k = \Gamma'$, and $\langle \Gamma_i, \Gamma_{i+1} \rangle$ is a planar linear morph, for each $i = 1, \dots, k-1$. A *linear morph* $\langle \Gamma_i, \Gamma_{i+1} \rangle$ is such that each vertex moves along a straight-line segment at uniform speed; that is, assuming that the morph happens between time $t = 0$ and time $t = 1$, the position of a vertex v at any time $t \in [0, 1]$ is $(1-t)\Gamma_i(v) + t\Gamma_{i+1}(v)$. The complexity of a morph $\mathcal{M}$ is then measured by the number of its *steps*, i.e., by the number of linear morphs it consists of.

A recent sequence of papers [3–6] culminated in a proof [2] that a planar morph between any two topologically-equivalent planar straight-line drawings of the same n-vertex planar graph can always be constructed consisting of $\Theta(n)$ steps. This bound is asymptotically optimal in the worst case, even for paths.

The question we study in this paper is whether morphs with sub-linear complexity can be constructed if a third dimension is allowed to be used. That is: Given two topologically-equivalent planar straight-line drawings Γ and Γ' of the same n-vertex planar graph G does a morph $\mathcal{M} = \langle \Gamma = \Gamma_1, \Gamma_2, \dots, \Gamma_k = \Gamma' \rangle$ exist such that: (i) for $i = 1, \dots, k$, the drawing Γ_i is a crossing-free straight-line 3D drawing of G, i.e., a straight-line drawing of G in $\mathbb{R}^3$ such that no two edges cross; (ii) for $i = 1, \dots, k-1$, the step $\langle \Gamma_i, \Gamma_{i+1} \rangle$ is a crossing-free linear morph, i.e., no two edges cross throughout the transformation; and (iii) $k = o(n)$? A morph $\mathcal{M}$ satisfying properties (i) and (ii) is a *crossing-free 3D morph*.

[1] Two planar drawings of a connected graph are *topologically equivalent* if they define the same clockwise order of the edges around each vertex and the same outer face.

[2] A *straight-line drawing* Γ of a graph G maps vertices to points in a Euclidean space and edges to open straight-line segments between the images of their end-vertices. We denote by $\Gamma(v)$ (by $\Gamma(G')$) the image of a vertex v (of a subgraph G' of G, resp.).

Our main result is a positive answer to the above question for trees. Namely, we prove that, for any two planar straight-line drawings Γ and Γ' of an n-vertex tree T, there is a crossing-free 3D morph with $O(\log n)$ steps between Γ and Γ'. More precisely the number of steps in the morph is linear in the *pathwidth* of T. Notably, our morphing algorithm works even if Γ and Γ' are not topologically equivalent, hence the use of a third dimension overcomes another important limitation of planar two-dimensional morphs. Our algorithm morphs both Γ and Γ' to an intermediate suitably-defined *canonical 3D drawing*; in order to do that, a root-to-leaf path H of T is moved to a vertical line and then the subtrees of T rooted at the children of the vertices in H are moved around that vertical line, thus resembling a pole dance, from which the title of the paper comes.

We also look at whether our result can be generalized to morphs of crossing-free straight-line 3D drawings of trees. That is, the drawings Γ and Γ' now live in $\mathbb{R}^3$, and the question is again whether a crossing-free 3D morph between Γ and Γ' exists with $o(n)$ steps. We prove that this is not the case: Two crossing-free straight-line 3D drawings of a path might require $\Omega(n)$ steps to be morphed one into the other. The matching upper bound can always be achieved: For any two crossing-free straight-line 3D drawings Γ and Γ' of the same n-vertex tree T there is a crossing-free 3D morph between Γ and Γ' with $O(n)$ steps.

The rest of the paper is organized as follows. In Sect. 2 we deal with crossing-free 3D morphs of 3D tree drawings. In Sect. 3 we show how to construct 2-step crossing-free 3D morphs between planar straight-line drawings of a path. In Sect. 4 we present our main result about crossing-free 3D morphs of planar tree drawings. Finally, in Sect. 5 we conclude and present some open problems.

Because of space limitations, some proofs are omitted or just sketched; they can be found in the full version of the paper.

2 Morphs of 3D Drawings of Trees

In this section we give a tight $\Theta(n)$ bound on the number of steps in a crossing-free 3D morph between two crossing-free straight-line 3D tree drawings.

Theorem 1. *For any two crossing-free straight-line 3D drawings Γ, Γ' of an n-vertex tree T, there exists a crossing-free 3D morph from Γ to Γ' that consists of $O(n)$ steps.*

Proof (sketch). The proof is by induction on n. The base case, in which $n = 1$, is trivial. If $n > 1$, then we remove a leaf v and its incident edge uv from T, Γ, and Γ'. This results in an $(n-1)$-vertex tree T' and two drawings Δ and Δ' of it. By induction, there is a crossing-free 3D morph between Δ and Δ'. We introduce v in such a morph so that it is arbitrarily close to u throughout the transformation; this significantly helps to avoid crossings in the morph. The number of steps is the one of the recursively constructed morph plus one initial step to bring v close to u, plus two final steps to bring v to its final position. □

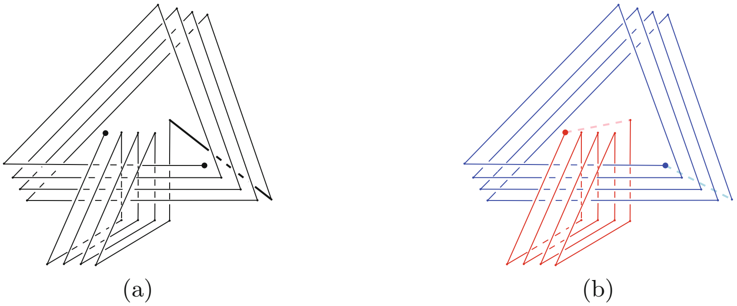


Fig. 1. Illustration for the proof of Theorem 2: (a) The drawing Γ of P, with $n = 26$; (b) the link obtained from Γ; the invisible edges are dashed.

Theorem 2. *There exist two crossing-free straight-line 3D drawings Γ, Γ' of an n-vertex path P such that any crossing-free 3D morph from Γ to Γ' consists of $\Omega(n)$ steps.*

Before proving Theorem 2, we review some definitions and facts from knot theory; refer, e.g., to the book by Adams [1]. A *knot* is an embedding of a circle S^1 in $\mathbb{R}^3$. A *link* is a collection of knots which do not intersect, but which may be linked together. For links of two knots, the (absolute value of the) *linking number* is an invariant that classifies links with respect to ambient isotopies. Intuitively, the linking number is the number of times that each knot winds around the other. The linking number is known to be invariant with respect to different projections of the same link [1]. Given a projection of the link, the linking number can be determined by orienting the two knots of the link, and for every crossing between the two knots in the projection adding +1 or −1 if rotating the understrand respectively clockwise or counterclockwise lines it up with the overstrand (taking into account the direction).

Proof (Theorem 2). The drawing Γ of P is defined as follows. Embed the first $\lfloor n/2 \rfloor$ edges of P in 3D as a spiral of monotonically decreasing height. Embed the rest of P as a same type of spiral affinely transformed so that it goes around one of the sides of the former spiral. See Fig. 1a. The drawing Γ' places the vertices of P in order along the unit parabola in the plane $y = 0$.

Cut the edge joining the two spirals (the bold edge in Fig. 1a). Removing an edge makes morphing easier so any lower bound would still apply. Now close the two open curves using two *invisible* edges to obtain a *link* of two knots; see Fig. 1b. It is easy to verify that the (absolute value of the) linking number of this link is $\Omega(n^2)$: indeed, determining it by the above procedure for the projection given by Fig. 1 results in the linking number being equal to the number of crossings between the two links in this projection. In the drawing Γ', each of the two halves of P (and their invisible edges) are separated by a plane and so their linking number is 0.

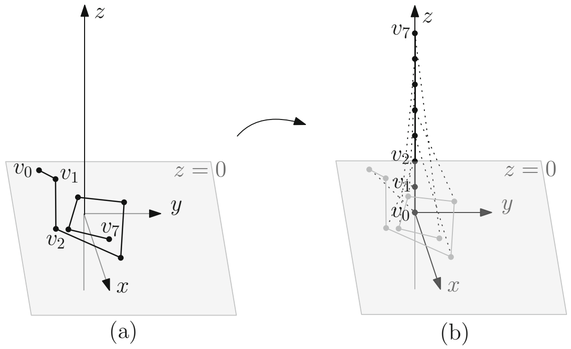
In a valid linear morph, the edges of P cannot cross each other, but they can cross invisible edges. However, during a linear morph between two straight-line

3D drawings of a graph G any two non-adjacent edges of G intersect $O(1)$ times. Thus each invisible edge can only be crossed $O(n)$ times during a linear morph. A single crossing can only change the linking number by 1. Therefore the linking number can only decrease by $O(n)$ in a single linear morph. □

3 Morphing Two Planar Drawings of a Path in 3D

In this section we show how to morph two planar straight-line drawings Γ and Γ' of an n-vertex path $P := (v_0, \ldots v_{n-1})$ into each other in two steps.

The *canonical 3D drawing* of P, denoted by $\mathcal{C}(P)$, is the crossing-free straight-line 3D drawing of P that maps each vertex v_i to the point $(0, 0, i) \in \mathbb{R}^3$, as shown in Fig. 2. We now prove the following.



(a)

(b)

Fig. 2. (a) A straight-line planar drawing Γ of an n-vertex path P and (b) a morph from Γ to $\mathcal{C}(P)$. The vertex trajectories are represented by dotted lines.

Theorem 3. *For any two planar straight-line drawings Γ and Γ' of an n-vertex path P, there exists a crossing-free 3D morph $\mathcal{M} = \langle \Gamma, \mathcal{C}(P), \Gamma' \rangle$ with 2 steps.*

Proof. It suffices to prove that the linear morph $\langle \Gamma, \mathcal{C}(P) \rangle$ is crossing-free, since the morph $\langle \mathcal{C}(P), \Gamma' \rangle$ is just the morph $\langle \Gamma', \mathcal{C}(P) \rangle$ played backwards.

Since $\langle \Gamma, \mathcal{C}(P) \rangle$ is linear, the speed at which the vertices of P move is uniform (though it might be different for different vertices). Thus the speed at which their projections on the z-axis move is uniform as well. Since v_i moves uniformly from $(x_i, y_i, 0)$ to $(0, 0, i)$, at any time during the motion (except at the time $t = 0$) we have $z(v_0) < z(v_1) < \ldots < z(v_{n-1})$. Therefore, in any intermediate drawing any edge (v_i, v_{i+1}) is separated from any other edge by the horizontal plane through one of its end-points. Hence no crossing happens during $\langle \Gamma, \mathcal{C}(P) \rangle$. □

4 Morphing Two Planar Drawings of a Tree in 3D

Let T be a tree with n vertices, arbitrarily rooted at any vertex. In this section we show that any two planar straight-line drawings of T can be morphed into one another by a crossing-free 3D morph with $O(\log n)$ steps (Theorem 4). Similarly to Sect. 3, we first define a canonical 3D drawing $\mathcal{C}(T)$ of T (see Sect. 4.1), and then show how to construct a crossing-free 3D morph from any planar straight-line drawing of T to $\mathcal{C}(T)$. We describe the morphing procedure in Sect. 4.2; then in Sect. 4.3 we present a procedure *Space*() that carries out the computations required by the morphing procedure; finally, in Sect. 4.4 we analyze the correctness and efficiency of both procedures.

Before proceeding, we introduce some necessary definitions and notation. By a *cone* we mean a straight circular cone induced by a ray rotated around a fixed vertical line (the *axis*) while keeping its origin fixed at a point (the *apex*) on this line. The *slope* $\phi(C)$ of a cone C, is the slope of the generating ray as determined in the vertical plane containing the ray. By a *cylinder* we always mean a straight cylinder having a horizontal circle as a base. Such cones or cylinders are uniquely determined, up to translations, respectively by their apex and slope or by their height and radius.

For a tree T, let $T(v)$ denote the subtree of T rooted at its vertex v. Also let $|T|$ denote the number of vertices in T. The *heavy-path decomposition* [15] of a tree T is defined as follows. For each non-leaf vertex v of T, let w be the child of v in T such that $|T(w)|$ is maximum (in case of a tie, we choose the child arbitrarily). Then (v, w) is a *heavy edge*; further, each child z of v different from w is a *light child* of v, and the edge (v, w) is a *light edge*. Connected components of heavy edges form paths, called *heavy paths*, which may have many incident light edges. Each path has a vertex, called the *head*, that is the closest vertex to the root of T. See Fig. 3 for an example. A *path tree* of T is a tree whose vertices correspond to heavy paths in T. The parent of a heavy path P in the path tree is the heavy path that contains the parent of the head of P. The root of the path tree is the heavy path containing the root of T. It is well-known [15] that the height of the path tree is $O(\log n)$. We denote by $H(T)$ the root of the path tree of T; let $v_0, \ldots, v_{k-1}$ be the ordered sequence of the vertices of $H(T)$, where v_0 is the root of T. For $i = 0, \ldots, k-1$, we let $v_i^0, \ldots, v_i^{t_i}$ be the light children of v_i in any order. Let $L(T) = u_0, u_1, \ldots, u_{l-1}$ be the sequence of the light children of $H(T)$ ordered so that: (i) any light child of a vertex v_j precedes any light child of a vertex v_i, if $i < j$; and (ii) the light child v_i^{j+1} of a vertex v_i precedes the light child v_i^j of v_i. When there is no ambiguity we refer to $H(T)$ and $L(T)$ simply as H and L, respectively.

4.1 Canonical 3D Drawing of a Tree

We define the *canonical 3D drawing* $\mathcal{C}(T)$ of a tree T as a straight-line 3D drawing of T that maps each vertex v of T to its *canonical position* $\mathcal{C}(v)$ defined as follows (see Fig. 3b). Note that our canonical drawing is equivalent to the *"standard"* straight-line upward drawing of a tree [7,9,10].

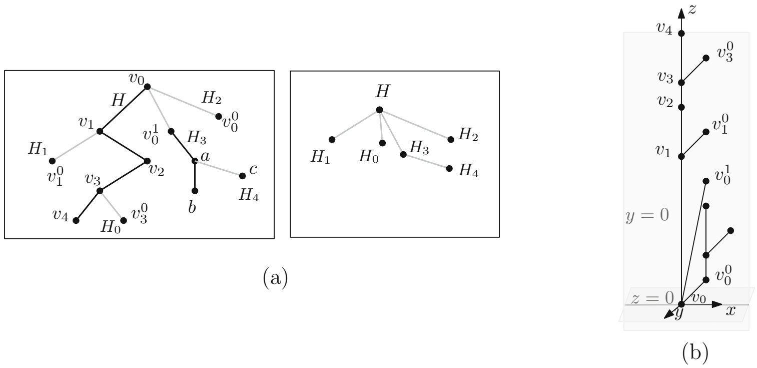


Fig. 3. (a) A tree T; (left) its heavy edges (bold lines) forming the heavy paths $H = H(T), H_0, \ldots, H_4$, and (right) the path tree of T; (b) $\mathcal{C}(T)$ for the tree T in (a).

First, we set $\mathcal{C}(v_0) = (0, 0, 0)$ for the root v_0 of T. Second, for each $i = 1, \ldots, k-1$, we set $\mathcal{C}(v_i) = (0, 0, z_{i-1} + |T(v_{i-1})| - |T(v_i)|)$, where z_{i-1} is the z-coordinate of $\mathcal{C}(v_{i-1})$. Third, for each $i = 1, \ldots, k-1$ and for each light child v_i^j of v_i, we determine $\mathcal{C}(v_i^j)$ as follows. If $j = 0$, we set $\mathcal{C}(v_i^j) = (1, 0, 1 + z_i)$, where z_i is the z-coordinate of $\mathcal{C}(v_i)$; otherwise, we set $\mathcal{C}(v_i^j) = (1, 0, z_i^{j-1} + |T(v_i^{j-1})|)$, where z_i^{j-1} is the z-coordinate of $\mathcal{C}(v_i^{j-1})$. Finally, in order to determine the canonical positions of the vertices in $T(v_i^j) \setminus \{v_i^j\}$, we recursively construct the canonical 3D drawing $\mathcal{C}(T(v_i^j))$ of $T(v_i^j)$, and translate all the vertices by the same vector so that v_i^j is sent to $\mathcal{C}(v_i^j)$.

Remark 1. *Notice that the canonical position $\mathcal{C}(v)$ of any vertex v of T is $(\mathrm{dpt}(v), 0, \mathrm{dfs}(v))$. Here $\mathrm{dpt}(v)$ is the depth, in the path tree of T, of the node that corresponds to the heavy path of T that contains v; and $\mathrm{dfs}(v)$ is the position of v in a depth-first search on T in which the children of any vertex are visited as follows: first visit the light children in reverse order with respect to L, and then visit the child incident to the heavy edge.*

The following lemma is a direct consequence of the construction of $\mathcal{C}(T)$.

Lemma 1. *The canonical 3D drawing $\mathcal{C}(T)$ of T lies on a rectangular grid in the plane $y = 0$, where the grid has height n and width equal to the height $h = O(\log n)$ of the path tree of T. Moreover, $\mathcal{C}(T)$ is on or above the line $z = x$.*

Remark 2. *In the above definition of the canonical 3D drawing $\mathcal{C}(T)$, instead of the heavy-path decomposition of T, we can use the decomposition based on the* Strahler number *of T, see [7] where the Strahler number is used under the name* rooted pathwidth *of T. With this change, the width of $\mathcal{C}(T)$ will be equal to the Strahler number of T, which is the instance-optimal width of an upward drawing of a tree [7]. Moreover, since the Strahler number is linear in the* pathwidth *of T,* *so is the width of $\mathcal{C}(T)$ defined this way. This is clearly not worse, and, for some instances, much better than the width given by the heavy-path decomposition.*

In the below description of the morph we use heavy paths, however we can use the paths given by Remark 2 instead, without any modification.

4.2 The Procedure *Canonize*($\boldsymbol{\Gamma}$)

Let $\Gamma = \Gamma(T)$ be a planar straight-line drawing of a tree T. Below we give a recursive procedure *Canonize*(Γ) that constructs a crossing-free 3D morph from Γ to the canonical 3D drawing $\mathcal{C}(T)$. We assume that Γ is enclosed in a disk of diameter 1 centered at $(0, 0, 0)$ in the plane $z = 0$, and that the root v_0 of T is placed at $(0, 0, 0)$ in Γ. This is not a loss of generality, up to a suitable modification of the reference system.

Step 1 (set the pole). The first step of the procedure *Canonize*(Γ) aims to construct a linear morph $\langle \Gamma, \Gamma_1 \rangle$, where Γ_1 is such that the heavy path $H = (v_0, \ldots, v_{k-1})$ of T lies on the vertical line through $\Gamma(v_0)$ and the subtrees of T rooted at the light children of each vertex v_i lie on the horizontal plane through v_i. More precisely, the vertices of T are placed in Γ_1 as follows. For $i = 0, \ldots, k-1$, place v_i at the point $\mathcal{C}(v_i)$. Every vertex that belongs to a subtree rooted at a light child of v_i is placed at a point such that its trajectory in the morph defines the same vector as the trajectory of v_i.[3] Below we refer to $\Gamma_1(H)$ as the *pole*. The pole will remain still throughout the rest of the morph.

Step 2 (lift). The aim of the second step of the procedure *Canonize*(Γ) is to construct a linear morph $\langle \Gamma_1, \Gamma_2 \rangle$, where Γ_2 is such that the drawings of any two subtrees $T(u_i)$ and $T(u_j)$ rooted at different light children u_i and u_j of vertices in H are vertically and horizontally separated. The separation between $\Gamma_2(T(u_i))$ and $\Gamma_2(T(u_j))$ is set to be large enough so that the recursively computed morphs *Canonize*$(\Gamma_2(T(u_i)))$ and *Canonize*$(\Gamma_2(T(u_j)))$ do not interfere with each other.

We describe how to construct Γ_2. As anticipated, $\Gamma_2(v_i) = \Gamma_1(v_i)$, for each vertex v_i in H. In order to determine the placement of the vertices not in H we use l cones $C^{\text{in}}_{u_0}$, $\ldots$,$C^{\text{in}}_{u_{l-1}}$ and l cones $C^{\text{out}}_{u_0}$, $\ldots$,$C^{\text{out}}_{u_{l-1}}$, namely one cone $C^{\text{in}}_{u_t}$ and one cone $C^{\text{out}}_{u_t}$ per vertex u_t in L. We also use, for each u_t, a cylinder *Space*$(\Gamma_2(T(u_t)))$ that bounds the volume used by *Canonize*$(\Gamma_2(T(u_t)))$. We defer the computation of these cones and cylinders to Sect. 4.3, and for now assume that they are already available. For each $t = 0, \ldots, l-1$ and for each $j = 0, \ldots, t-1$, assume that $\Gamma_2(T(u_j))$ has been computed already – this is indeed the case when $t = 0$. Let $\mathcal{P}_t$ be the horizontal plane $z = |T| - 1 + \sum_{j=0}^{t-1} h(u_j)$, where $h(u_j)$ is the height of the cylinder *Space*$(\Gamma_2(T(u_j)))$. The drawing Γ_2 maps the subtree $T(u_t)$ to the plane $\mathcal{P}_t$, just outside the cone $C^{\text{in}}_{u_t}$ and just inside the

[3] Since the morph $\langle \Gamma, \Gamma_1 \rangle$ is linear, the trajectory of any vertex v is simply the line segment connecting the positions of v in Γ and in Γ_1. To define a vector, we orient the segment towards the position of v in Γ_1.

cone $C^{\text{out}}_{u_t}$. See Fig. 4. We proceed with the formal definition of Γ_2. Let v be any vertex of $T(u_t)$ and let (v_x, v_y, v_z) be the coordinates of $\Gamma_1(v)$. Then $\Gamma_2(v)$ is the point $(v_x \frac{r_t}{r}, v_y \frac{r_t}{r}, z_t)$. Here z_t is the height of the plane $\mathcal{P}_t$, r_t is the radius of the section of $C^{\text{in}}_{u_t}$ by the plane $\mathcal{P}_t$, and r is the distance from $\Gamma_1(v_i)$ to its closest point of the drawing $\Gamma_1(T(u_t))$, where v_i is the parent of u_t. See Fig. 4. Note that the latter closest point can be a point on an edge.

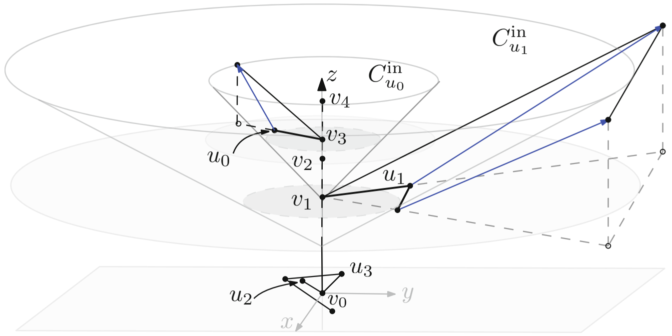


Fig. 4. The vertices v_0, v_1, v_2, v_3, v_4 are in the heavy path H of T. The lower gray disk has its center at v_1 and has radius equal to the distance from $\Gamma_1(v_i)$ to its closest point in $\Gamma_1(T(u_1))$. Blue arrows show the mapping of vertices in subtrees $T(u_0)$ and $T(u_1)$.

Step 3 (recurse). For each $u_t \in L$, we make a recursive call *Canonize*$(\Gamma_2(T(u_t)))$. The resulting morphs are combined into a unique morph $\langle \Gamma_2, \dots, \Gamma_3 \rangle$, whose number of steps is equal to the maximum number of steps in any of the recursively computed morphs. Indeed, the first step of $\langle \Gamma_2, \dots, \Gamma_3 \rangle$ consists of the first steps of all the recursively computed morphs that have at least one step; the second step of $\langle \Gamma_2, \dots, \Gamma_3 \rangle$ consists of the second steps of all the t recursively computed morphs that have at least two steps; and so on.

Step 4 (rotate, rotate, rotate). The next morph transforms Γ_3 into a drawing Γ_4 such that each vertex $u_t \in L$ is mapped to the intersection of the cone $C^{\text{in}}_{u_t}$, the planes $y = 0$, $\mathcal{P}_t$, and the half-space $x > 0$. Note that going from Γ_3 to Γ_4 in one linear crossing-free 3D morph is not always possible. Refer to Lemma 2 for the implementation of the morph from Γ_3 to Γ_4 in $O(1)$ steps. After Step 4 the whole drawing lies on the plane $y = 0$.

Step 5 (go down). This step consists of a single linear morph $\langle \Gamma_4, \Gamma_5 \rangle$, where Γ_5 is defined as follows. For every vertex v_i in H, $\Gamma_5(v_i) = \Gamma_4(v_i)$; further, for every vertex $u_t \in L$, all the vertices of $T(u_t)$ have the same x- and y-coordinates in Γ_5 as in Γ_4, however their z-coordinate is decreased by the same amount so that u_t lies on the horizontal plane through $\mathcal{C}(u_t)$.

Step 6 (go left). The final part of our morphing procedure consists of a single linear morph $\langle \Gamma_5, \Gamma_6 \rangle$, where Γ_6 is the canonical 3D drawing $\mathcal{C}(T)$ of T. Note that this linear morph only moves the vertices horizontally.

4.3 The Procedure *Space*(Γ)

In this section we give a procedure to compute the cylinders and the cones which are necessary for Steps 2 and 4 of the procedure *Canonize*(Γ).

We fix a constant $c \in \mathbb{R}$ with $c > 1$, which we consider global to the procedure *Canonize*(Γ) and its recursive calls; below we refer to c as the *global constant*. The global constant c will help us to define the cones so that Step 4 of *Canonize*(Γ) can be realized with $O(1)$ linear morphs, see Lemma 2.

The procedure *Space*(Γ) returns a cylinder that encloses all the intermediate drawings of the morph determined by *Canonize*(Γ). At the same time, *Space*(Γ) determines the cones $C^{\text{in}}_{u_t}$ and $C^{\text{out}}_{u_t}$ for every vertex $u_t \in L$.

We now describe *Space*(Γ). Let Γ_1 be the result of the application of Step 1 of *Canonize*(Γ). Figure 5 illustrates our description.

If T is a path, i.e., $T = H$, return the cylinder of height $|T| - 1$ and radius 1. In particular, if T is a single vertex, return the disk of radius 1. Otherwise, construct the cylinder and the cones in the following fashion:

- Set the current cone C to be an infinite cone of slope 1. The apex of C is determined as follows: starting with the apex being at the highest point of the pole, slide C vertically downwards until it touches the drawing $\Gamma_1(T(u_0))$. That is, the apex of C is at the lowest possible position on the pole such that the whole drawing $\Gamma_1(T(u_0))$ is outside of C. See Fig. 5a.
- Set the current height h to be $|T| - 1$.
- Iterate through the light children of H in the order as they appear in L. For every u_t in L:
  - Set $C^{\text{in}}_{u_t}$ to be the current cone C.
  - Add the height of *Space*($\Gamma_2(T(u_t))$) to the current height h.
  - Let C' be the cone with the same apex as C and with a slope defined so that the drawing $\Gamma_1(T(u_t))$ is in-between C and C', and C is *well-separated* from C' with the global constant c. That is, $\phi(C') = \min(\phi(C)/Sp(u_t, \Gamma_1), \phi(C)/c)$, where $Sp(u_t, \Gamma_1)$ is the spread of the drawing $\Gamma_1(T(u_t))$ with respect to the parent v_i of u_t in H. Namely $Sp(u_t, \Gamma_1)$ is the ratio between the outer and the inner radius of the minimum annulus centered at v_i and enclosing the drawing $\Gamma_1(T(u_t))$. See Fig. 5a.
  - Let $\mathcal{S}_t$ be the cylinder *Space*($\Gamma_2(T(u_t))$) translated so that the center of its lower base is at the point $\Gamma_2(u_t)$.
  - Decrease $\phi(C')$ so that C' encloses the entire cylinder $\mathcal{S}_t$.
  - Set $C^{\text{out}}_{u_t}$ to be the cone C'.
  - If u_t is not the last element of L (i.e., $t < l - 1$), then let $u_t = v_i^j$ and define an auxiliary cone $\tilde{C}$ as follows. The apex of $\tilde{C}$ is at $\Gamma_1(v_x)$ where v_x is the parent of u_{t+1}; note that $v_x = v_i$ iff $j > 0$. The slope of $\tilde{C}$ is the

maximum slope that satisfies the following requirement: (i) the slope of $\tilde{C}$ is at most the slope of C'. In addition, only for the case when $v_x = v_i$, we require: (ii) in the closed half space $z \leq h$, the portion of $\tilde{C}$ encloses the portion of C'. See Fig. 5b. Update the cone C to be the lowest vertical translate of $\tilde{C}$ so that $\Gamma_1(T(u_{t+1}))$ is still outside the cone.

- Return the cylinder of height h (the current height), and radius equal to the radius of the section of the current cone C cut by the plane $z = h$.

4.4 Correctness of the Morphing Procedure

In this section, we analyze the correctness and the efficiency of the procedure *Canonize*(Γ) (see Theorem 4) and we give the details of Step 4 (see Lemma 2).

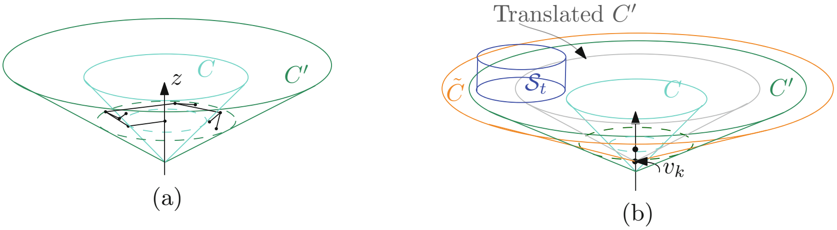


Fig. 5. Illustration for *Space*(Γ): (a) construction of C and C'; (b) construction of $\tilde{C}$.

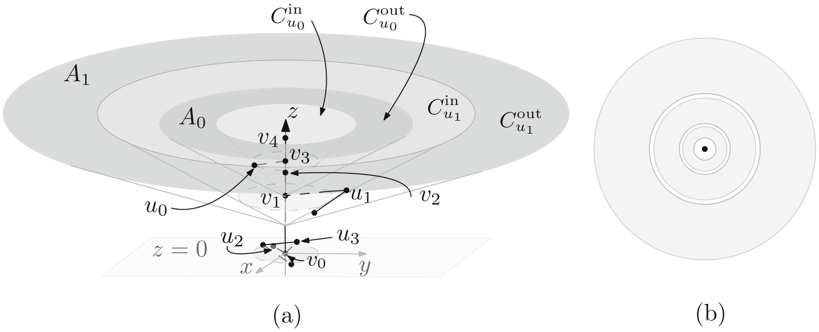


Fig. 6. (a) Annuli for the subtrees rooted at u_0 and u_1; (b) top view of the annuli.

Lemma 2. *Step 4 of the procedure Canonize(Γ) can be realized as a crossing-free 3D morph whose number of steps is bounded from above by a constant that depends on the global constant c.*

Proof. Let A_t be the annulus formed by the section of $C^{\text{in}}_{u_t}$ and $C^{\text{out}}_{u_t}$ cut by the plane $\mathcal{P}_t$. See Fig. 6. The morph performed in Step 4 consists of a sequence of linear morphs; in each of these morphs all the vertices of $T(u_t)$ are translated by the same vector. This is done so that u_t stays in A_t during the whole Step 4. Thus, the trajectory of u_t during Step 4 defines a polygon inscribed in A_t. Since the ratio between the outer and the inner radius of A_t is at least the global constant c, we can inscribe a regular $O(1)$-gon in A_t, and the trajectory of u_t can be defined so that it follows this $O(1)$-gon plus at most one extra line segment.

We now prove that since each $u_t \in L$ stays in A_t, all the steps of the above morph are crossing-free. Recall that at any moment during the morph, the drawing of $T(u_t)$ is a translation of the canonical 3D drawing $\mathcal{C}(T(u_t))$. By Lemma 1, the space below the line of slope 1 passing through u_t in plane $y = 0$ does not contain any point of $\mathcal{C}(T(u_t))$. Since the slope of $C^{\text{out}}_{u_t}$ is at most 1, the drawing of $T(u_t)$ is enclosed in $C^{\text{out}}_{u_t}$ as long as u_t is in A_t. By conditions (i) and (ii) of $\mathit{Space}(\Gamma)$, the cone $C^{\text{in}}_{u_{t+1}}$ encloses $C^{\text{out}}_{u_t}$ in the closed half-space above $\mathcal{P}_t$. Hence the edge connecting u_{t+1} to the pole never touches $C^{\text{out}}_{u_t}$ above $\mathcal{P}_t$. □

Theorem 4. *For any two plane straight-line drawings Γ, Γ' of an n-vertex tree T, there exists a crossing-free 3D morph from Γ to Γ' with $O(\log n)$ steps.*

Proof (sketch). A 3D morph from Γ to Γ' can be constructed as the concatenation of $\mathit{Canonize}(\Gamma)$ with the reverse of $\mathit{Canonize}(\Gamma')$. Hence, it suffices to prove that $\mathit{Canonize}(\Gamma)$ is a crossing-free 3D morph with $O(\log n)$ steps.

It is easy to see that Steps 1, 5, and 6 of $\mathit{Canonize}(\Gamma)$ are crossing-free linear morphs. The proof that Step 2 is a crossing-free linear morph is more involved. In particular, for any two light children u_s and u_t with $s < t$ of the same vertex v_i of H, the occurrence of a crossing between the edge $v_i u_s$ and an edge of $T(u_t)$ during Step 2 can be ruled out by arguing that the same two edges would also cross in Γ_1; this argument exploits the uniformity of the speed in a linear morph and that the horizontal component of the morph of Step 2 is a uniform scaling. Lemma 2 ensures that Step 4 is a crossing-free 3D morph with $O(1)$ steps. Thus, Steps 1, 2, 4, 5, and 6 require a total of $O(1)$ steps. Since the number of morphing steps of Step 3 of $\mathit{Canonize}(\Gamma)$ is equal to the maximum number of steps of any recursively computed morph and since, by definition of heavy path, each tree $T(u_t)$ for which a recursive call $\mathit{Canonize}(\Gamma_2(T(u_t)))$ is made has at most $n/2$ vertices, it follows that $\mathit{Canonize}(\Gamma)$ requires $O(\log n)$ steps. □

5 Conclusions and Open Problems

In this paper we studied crossing-free 3D morphs of tree drawings. We proved that, for any two planar straight-line drawings of the same n-vertex tree, there is a crossing-free 3D morph between them which consists of $O(\log n)$ steps.

This result gives rise to two natural questions. First, is it possible to bring our logarithmic upper bound down to constant? In this paper we gave a positive answer to this question for paths. In fact our algorithm to morph planar straight-line tree drawings has a number of steps which is linear in the pathwidth of the tree (see Remark 2), thus for example it is constant for caterpillars. Second, does a crossing-free 3D morph exist with $o(n)$ steps for any two planar straight-line drawings of the same n-vertex planar graph? The question is interesting to us even for subclasses of planar graphs, like outerplanar graphs and planar 3-trees.

We also proved that any two crossing-free straight-line 3D drawings of an n-vertex tree can be morphed into each other in $O(n)$ steps; such a bound is asymptotically optimal in the worst case. An easy extension of our results to graphs containing cycles seems unlikely. Indeed, the existence of a deterministic algorithm to construct a crossing-free 3D morph with a polynomial number of steps between two crossing-free straight-line 3D drawings of a cycle would imply that the unknot recognition problem is polynomial-time solvable. The *unknot recognition* problem asks whether a given knot is equivalent to a circle in the plane under an ambient isotopy. This problem has been the subject of investigation for decades; it is known to be in NP [13] and in co-NP [14], however determining whether it is in P has been an elusive goal so far.

Acknowledgments. We thank Therese Biedl for pointing out Remark 2. The research for this paper started during the Intensive Research Program in Discrete, Combinatorial and Computational Geometry, which took place in Barcelona, April-June 2018. We thank Vera Sacristán and Rodrigo Silveira for a wonderful organization and all the participants for interesting discussions.

References

1. Adams, C.C.: The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots. American Mathematical Society, Providence (2004)
2. Alamdari, S., et al.: How to morph planar graph drawings. SIAM J. Comput. **46**(2), 824–852 (2017)
3. Alamdari, S., et al.: Morphing planar graph drawings with a polynomial number of steps. In: Khanna, S. (ed.) 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2013), pp. 1656–1667. SIAM (2013)
4. Angelini, P., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M., Roselli, V.: Morphing planar graph drawings optimally. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8572, pp. 126–137. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43948-7_11
5. Angelini, P., Frati, F., Patrignani, M., Roselli, V.: Morphing planar graph drawings efficiently. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 49–60. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03841-4_5
6. Barrera-Cruz, F., Haxell, P., Lubiw, A.: Morphing planar graph drawings with unidirectional moves. In: Mexican Conference on Discrete Mathematics and Computational Geometry, pp. 57–65 (2013). http://arxiv.org/abs/1411.6185
7. Biedl, T.: Optimum-width upward drawings of trees. arXiv preprint arXiv:1506.02096 (2015)
8. Cairns, S.S.: Deformations of plane rectilinear complexes. Am. Math. Mon. **51**(5), 247–252 (1944)
9. Chan, T.M.: Tree drawings revisited. arXiv preprint arXiv:1803.07185 (2018)
10. Crescenzi, P., Di Battista, G., Piperno, A.: A note on optimal area algorithms for upward drawings of binary trees. Comput. Geom. **2**(4), 187–200 (1992)

11. Floater, M.S., Gotsman, C.: How to morph tilings injectively. J Comput. Appl. Math. **101**(1–2), 117–129 (1999)
12. Gotsman, C., Surazhsky, V.: Guaranteed intersection-free polygon morphing. Comput. Graph. **25**(1), 67–75 (2001)
13. Hass, J., Lagarias, J.C., Pippenger, N.: The computational complexity of knot and link problems. J. ACM **46**(2), 185–211 (1999)
14. Lackenby, M.: The efficient certification of knottedness and Thurston norm. CoRR, abs/1604.00290 (2016)
15. Sleator, D.D., Tarjan, R.E.: A data structure for dynamic trees. In: Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing, pp. 114–122 (1981)
16. Thomassen, C.: Deformations of plane graphs. J. Comb. Theory, Ser. B **34**(3), 244–257 (1983)