Morphing Tree Drawings in a Small 3D Grid

Aleksandra Istomina(✉), Elena Arseneva, and Rahul Gangopadhyay

Saint-Petersburg University, Saint Petersburg, Russia
st062510@student.spbu.ru, e.arseneva@spbu.ru, rahulg@iiitd.ac.in

Abstract. We study crossing-free grid morphs for planar tree drawings using the third dimension. A morph consists of morphing steps, where vertices move simultaneously along straight-line trajectories at constant speeds. There is a crossing-free morph between two drawings of an n-vertex planar graph G with $\mathcal{O}(n)$ morphing steps, and using the third dimension the number of steps can be reduced to $\mathcal{O}(\log n)$ for an n-vertex tree [Arseneva et al. 2019]. However, these morphs do not bound one practical parameter, the resolution. Can the number of steps be reduced substantially by using the third dimension while keeping the resolution bounded throughout the morph? We present a 3D crossing-free morph between two planar grid drawings of an n-vertex tree in $\mathcal{O}(\sqrt{n} \log n)$ morphing steps. Each intermediate drawing lies in a $3D$ grid of polynomial volume.

Keywords: morphing grid drawings · bounded resolution · 3D morphing

1 Introduction

Given an n-vertex graph G, a *morph* between two drawings (i.e., embeddings in $\mathbb{R}^d$) of G is a continuous transformation from one drawing to the other through a family of intermediate drawings. One is interested in well-behaved morphs, i.e., those that preserve essential properties of the drawing at any moment. Usually, this property is that the drawing is *crossing-free*; such morphs are called *crossing-free* morphs. This concept finds applications in multiple domains: animation, modeling, and computer graphics, etc. A drawing of G is a *straight-line drawing* if it maps each vertex of G to a point in $\mathbb{R}^d$ and each edge of G to the line segment whose endpoints correspond to the endpoints of this edge. In this work, we focus on the case of drawings in the Euclidean plane ($d = 2$) and $3D$ drawings ($d = 3$); a non-crossing drawing of a graph in $\mathbb{R}^2$ is called *planar*.

There is an interest in studying crossing-free morphs of straight-line drawings, where vertex trajectories are simple, in particular, *linear morphs*. A linear morph transforms one straight-line drawing Γ of a graph G to another such drawing Γ' through a sequence of straight-line drawings; each *morphing steps* or *step* is a linear interpolation between two consecutive drawings in that sequence.

P. Mutzel et al. (Eds.): WALCOM 2022, LNCS 13174, pp. 85–96, 2022.
https://doi.org/10.1007/978-3-030-96731-4_8

That is, during each morphing step each vertex of G moves along a straight-line segment at a constant speed. A linear morph is said to be *unidirectional* if all vertices move along parallel lines in the same direction. Alamdari et al. [1] showed that for any two topologically equivalent planar drawings of a graph G, there is a linear $2D$ morph that transforms one drawing to the other in $\Theta(n)$ steps. This bound is asymptotically optimal in the worst case even when the graph G is a path. A natural further question is how the situation changes when we involve the third dimension. For general 3D graph drawings the problem seems challenging since it is tightly connected to *unknot recognition* problem. If both the initial and the final drawing are planar and the given graph is a tree, then $\mathcal{O}(\log n)$ steps suffice [2]. In both algorithmic results [1,2], the intermediate steps use infinitesimal or very small distances, as compared to distances in the input drawings. This may blow up the space requirements and affect the aesthetical aspect. This raises a demand for morphing algorithms that operate on a small grid, i.e., of size that is polynomial in the size of the graph and parameters of the input drawings. All the intermediate drawings are then restricted to be *grid drawings*, where vertices map to vertices of the grid. Two crucial parameters of a straight-line grid drawing are: the area (or volume for the 3D case) of the required grid, and the *resolution*, that is the ratio between the maximum edge length and the minimum edge-edge distance. If the grid area (or volume) is polynomially bounded, then so is the resolution [3].

Very recently Barrera-Cruz et al. [3] gave an algorithm that linearly morphs between two planar straight-line grid drawings Γ and Γ' of an n-vertex rooted tree in $\mathcal{O}(n)$ steps while each intermediate drawing is also a planar straight-line drawing in a bounded grid. In particular, the maximum grid length and width are respectively $\mathcal{O}(D^3 n \cdot L)$ and $\mathcal{O}(D^3 n \cdot W)$, where $L = max\{l(\Gamma), l(\Gamma')\}$, $W = max\{w(\Gamma), w(\Gamma')\}$ and $D = max\{L, W\}$, $l(\Gamma)$ and $w(\Gamma)$ are the *length* and the *width* of the drawing Γ respectively. Note that D is $\Omega(\sqrt{n})$.

Let Γ and Γ' be two planar straight-line drawings of an n-vertex tree T. Throughout this paper, a morph $\mathcal{M} = \langle \Gamma_1, \Gamma_2, \ldots, \Gamma_k \rangle$ of T is a sequence of 3D straight-line drawings of T such that $\Gamma_1 = \Gamma, \Gamma_k = \Gamma'$ are the initial and the final drawings, and each $\langle \Gamma_i, \Gamma_{i+1} \rangle$ is a linear morph. Here we study the problem of morphing one straight-line grid drawing Γ to another such drawing Γ' in *sublinear* number of steps using the third dimension such that the resolutions of the intermediate drawings are bounded. We morph the initial planar drawing of tree T to its $3D$ canonical drawing $\mathcal{C}(T)$ and then analogously morph $\mathcal{C}(T)$ to the final planar drawing. Effectively we solve the same problem as in [2], but with the additional restriction that all drawings throughout the algorithm lie in a small grid. We give an algorithm that requires $\mathcal{O}(\sqrt{n} \log n)$ steps. All the intermediate drawings require a $3D$ grid of length $\mathcal{O}(d^3(\Gamma) \cdot \log n)$, width $\mathcal{O}(d^3(\Gamma) \cdot \log n)$ and height $\mathcal{O}(n)$, where $d(\Gamma) = \max(d(\Gamma), d(\Gamma'))$. During the procedure, we use some known techniques, e.g., canonical drawing [2] and "Pinwheel" rotation [3] combined with several new ideas.

In Sect. 2, we introduce the definitions that are used in the paper. After introducing the necessary definitions and preliminaries in Sect. 2, we describe the tools that are the building blocks of our algorithm: stretching, mapping around the pole, rotating and shrinking subtrees (See Sect. 3). In Sect. 4, we introduce a technique of lifting paths such that the vertices on the path along with their subtrees go to the respective canonical positions and the drawing remains crossing-free. The morphing algorithm in Sect. 4 splits the given tree into disjoint paths that are lifted one by one in specific order. Since lifting each path takes constant number of steps, in the worst case this algorithm takes $\mathcal{O}(n)$ steps to lift a tree. In Sect. 5, we show how to lift a set of edges of the given tree simultaneously. This is used in the second morphing algorithm, that lifts the tree by lifting disjoint sets of its edges one after another. This algorithm takes $\mathcal{O}(h)$ steps to lift a tree of height h. We then combine two algorithms in Sect. 6 to produce the final algorithm that uses $o(n)$ morphing steps. It first lifts all paths of T of length at most $\sqrt{n}$ using the algorithm of Sect. 5. Since the total number of remaining paths is less than $\sqrt{n}$, we lift them one after another by using the algorithm of Sect. 4. The full version[1] of the paper contains detailed proofs and descriptions which are omitted here due to space constraints.

2 Preliminaries and Definitions

Tree Drawings. For a rooted tree T, let $r(T)$ be its root, and $T(v)$ be the subtree of T rooted at a vertex v of T. Let $E(T)$, $V(T)$ and $|T|$ denote respectively, the set of edges, the set of vertices, and the number of vertices of T. In a *straight-line drawing* of T, each vertex is mapped to a point in $\mathbb{R}^d$ and each edge is mapped to a straight-line segment connecting its end-points. A $3D$- (respectively, a $2D$-) *grid drawing* of T is a straight-line drawing where each vertex is mapped to a point with integer coordinates in $\mathbb{R}^3$ (respectively, $\mathbb{R}^2$). A drawing of T is said to be *crossing-free* if images of no two edges intersect except, possibly, at common end-points. A crossing-free $2D$-grid drawing is called a *planar grid drawing*. For a crossing-free drawing Γ, let $B(\Gamma(v), r)$ denote the open disc of radius r in the horizontal plane centered at the image $\Gamma(v)$ of v. By the *projection*, denoted by $pr()$, we mean the vertical projection to the horizontal plane passing through the origin. Let $l(\Gamma)$, $w(\Gamma)$ and $h(\Gamma)$ respectively denote the *length*, *width* and *height* of the $3D$ drawing Γ of T, i.e., the maximum absolute difference between the x-, y- and z-coordinates of vertices in Γ. Let $d(\Gamma)$ denote the *diameter* of Γ, defined as the ceiling of the maximum pairwise (Euclidean) distance between its vertices. Note that $d(\Gamma)$ estimates the space required by Γ since $M \leq d(\Gamma) \leq \sqrt{3}M$, where $M = \max(l(\Gamma), w(\Gamma), h(\Gamma))$. Let $dist_\Gamma(v, e)$ (resp., $dist_\Gamma(v, u)$) be the distance between $\Gamma(v)$ and $\Gamma(e)$ (resp., between $\Gamma(v)$ and $\Gamma(u)$), where u, v are vertices of T and e is an edge of T. For a grid drawing Γ, we define the *resolution* of Γ as the ratio of the distances between the farthest and closest pairs of geometric objects of Γ (images of tree vertices and edges).

For any vertex v and edge e not incident to v in a crossing-free grid drawing Γ of T, $dist(v, e) \geq \frac{1}{d(\Gamma)}$. In a $3D$ grid drawing Γ of T, the distance $dist(e_1, e_2) \geq \frac{1}{2\sqrt{3}\,(d(\Gamma))^2}$ for a pair of non-adjacent edges e_1, e_2. This implies that $2D$ and

[1] arXiv:2106.04289.

$3D$ crossing-free grid drawings of T have polynomially bounded resolution. For a point $p = (p_x, p_y, p_z)$, we denote by YZ_p, XZ_p, XY_p planes $x = p_x, y = p_y, z = p_z$ respectively. Analogously, XZ_p^+ (resp., XZ_p^-) denotes the vertical half-plane $\{(x, y, z) : y = p_y, x \geq p_x(resp., x \leq p_x)\}$ and YZ_p^+ (resp., YZ_p^-) the half-plane $\{(x, y, z) : x = p_x, y \geq p_y(resp., y \leq p_y)\}$.

Path Decomposition. $\mathcal{P}$ of a tree T is a decomposition of its edges into a set of edge-disjoint paths as follows. Choose some root-to-leaf path in T and store it in the set $\mathcal{P}$ which is empty at the beginning. Remove the edges of this path from T. It may disconnect the tree; recurse on the remaining connected components while there are edges. In the end, $\mathcal{P}$ contains disjoint paths whose union is $E(T)$. The depth $dpt(v)$ of a vertex v in T is the length of the path from $r(T)$ to v. *Head* of a path P, denoted as $head(P)$, is the vertex $x \in P$ with the minimum depth in tree T. Let the *internal vertices* of path P be all vertices of P except $head(P)$. Any path decomposition $\mathcal{P}$ of T induces a linear order of the paths: path P' succeeds P, i.e., $P' \succ P$, if and only if P' is deleted before P during the construction of $\mathcal{P}$. Note that the subtree of each internal vertex of a path P is a subset of the union of the paths that precede P.

In the *long-path decomposition* [4] $\mathcal{L}(T)$, the path chosen in every iteration is the longest root-to-leaf path (ties are broken arbitrarily). Let $\mathcal{L} = \{L_1, \ldots, L_m\}$ be the ordered set of paths of a long-path decomposition of T. For $i < j$, $|L_i| \leq |L_j|$.

In the *heavy-rooted-pathwidth decomposition* $\mathcal{H}(T)$ (see, e.g., [2]), of a tree T, the root-to-leaf path chosen in every iteration maximizes the rooted pathwidth, $rpw(T)$.$rpw(T)$ is defined recursively: for each leave v of T: $rpw(\{v\}) = 1$; for each internal vertex u and its children $v_1, \ldots, v_k$ we have $rpw(T(u)) = max(rpw(T(v_i))), 1 \leq i \leq k$ if $rpw(T(v_i))$ are not all equal, and $rpw(T(u)) = rpw(T(v_1)) + 1$ in the other case. It is known [5] that for a tree T with n vertices $rpw(T) = \mathcal{O}(\log n)$. Figure 1a and 1b show respectively the heavy-rooted-pathwidth and the long-path decomposition of a tree where heavy paths and long paths are shown in different colors.

Canonical 3D drawing $\mathcal{C}(T)$ of a tree T [2] is the crossing-free straight-line 3D drawing of T that maps each vertex v of T to its *canonical position* $\mathcal{C}(v)$ determined by the heavy-rooted pathwidth decomposition. We later use the fact that $\mathcal{C}(T)$ lies in XZ_0^+ inside a bounding box of height $|T|$ and width $rpw(T)$. For any vertex v of T, the *relative canonical drawing* $\mathcal{C}_{T_v}$ of $T(v)$ is the drawing of $T(v)$ obtained by cropping $\mathcal{C}(T)$ and translating the obtained drawing of $T(v)$ so that v is mapped to the origin. Since tree T never changes throughout our algorithm, we refer to $rpw(T)$ as to rpw.

3 Tools for Morphing Algorithms

We define stretching, mapping, rotation and shrinking of subtrees in this section. Each of these are fundamental tools used in our algorithm.

Stretching with a Constant $\mathcal{S}_1$. Let the drawing Γ lie in the XY_0 plane. During *stretching morph* $\langle \Gamma, \Gamma_1 \rangle$ each coordinate of each vertex in Γ is multiplied by a common positive integer constant $\mathcal{S}_1$ to obtain Γ_1. Thereby, it is a linear morph that "stretches" the vertices apart. Stretching morph is crossing-free.

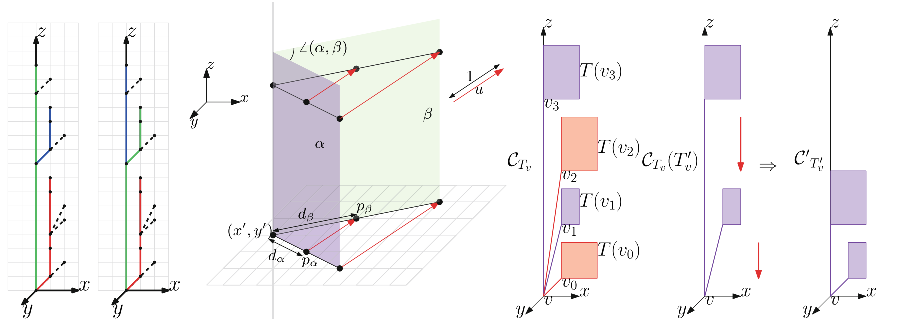

Fig. 1. Canonical drawing of a tree with (a) heavy and (b) long paths, where paths are colored with different colors, paths that consist of one edge are dashed. (c) The mapping morph, half-planes α, β sharing a common pole through point (x', y') and their vector of mapping. (d) The Shrinking morph when $l = 4$.

Lemma 1. *For any pair v_i, v_j of vertices disks $B(\Gamma_1(v_i), \frac{\mathcal{S}_1}{2})$ and $B(\Gamma_1(v_j), \frac{\mathcal{S}_1}{2})$ do not cross in the XY_0 plane. For a vertex v_i disk $B(\Gamma_1(v_i), \frac{\mathcal{S}_1}{2 \cdot d(\Gamma)})$ does not enclose any other vertices or any part of edges non-incident to v_i in Γ_1. For every vertex v and every edge $e = (v, u)$ in Γ_1 there is lattice point z such that $z \in e$ and $z \in B(\Gamma_1(v_i), d(\Gamma))$.*

Mapping Around a Pole. Let the *pole through* (x', y') be the vertical line in $3D$ through a point $(x', y', 0)$. Let α, β be vertical half-planes containing the pole l through a point with integer coordinates. Suppose $\angle(\alpha, \beta) \notin \{0, \pi\}$ and α, β contain infinitely many points with integer coordinates. Mapping *around the pole* l is a morphing step to obtain a drawing Γ' which lies in β from Γ which lies in α. Each vertex moves along a horizontal vector between α and β. The direction of this vector is common for all vertices of Γ and is defined by α and β. Let us fix a horizontal plane h passing through the point $(0, 0, b)$ where b is an integer. Let p_α, p_β be points that lie on $h \cap \alpha$ and $h \cap \beta$, respectively; such that $dist(l, p_\alpha) = d_\alpha$ and $dist(l, p_\beta) = d_\beta$ be the minimum non-zero distances from the l to the integer points lying in $h \cap \alpha$ and $h \cap \beta$. The *vector of mapping* u is defined as $\frac{p_\beta - p_\alpha}{|p_\beta - p_\alpha|}$. Mapping is an unidirectional morph since all vertices of Γ move along the vectors parallel to the vector of mapping till they reach the half-plane β, see Fig. 1c. Since mapping is comprised of rotation and stretching in horizontal direction, it is a crossing- free morph that preserves grid drawings. Throughout the paper, we denote by *rotation* a mapping when α, β are half-planes of planes parallel to XZ_0, YZ_0 respectively or vice-versa. Similarly, we define mapping around horizontal pole, i.e., a pole parallel to the X-axis.

Rotating Horizontal Plane. Let $\Gamma_0(T(v))$ be the canonical drawing of a subtree $T(v)$ on the horizontal plane XY_v obtained by rotating the relative canonical drawing $\mathcal{C}_{T_v}$ around the horizontal pole through v. Let

$\Gamma_1(T(v)), \Gamma_2(T(v)), \Gamma_3(T(v))$ be the drawings obtained from $\Gamma_0(T(v))$ by rotating the horizontal plane around the point $\Gamma(v)$ by the angles $\frac{\pi}{2}, \pi, \frac{3\pi}{2}$, respectively. In the Appendix we show that the drawing $\Gamma_i(T(v))$ can be obtained from the drawing $\Gamma_{i-1}(T(v))$ in one morphing step—*rotating step*—using a lemma from [3].

Shrinking Lifted Subtrees. Let v be a vertex of T. Assume that the image $\Gamma(T(v))$ of subtree $T(v)$ is $\mathcal{C}_{T_v}$, in particular, it lies in $h = XZ_v^+$. Let $\mathcal{C} = \{v_1, \ldots, v_l\}$ be sequence of children of v, ordered according to their z-coordinates in $\mathcal{C}_{T_v}$. Let $\mathcal{C}' = \{v_{i_1}, \ldots, v_{i_k}\}$ be subsequence of $\mathcal{C}$. Let us consider the new subtree $T'(v)$ which is obtained by deleting the vertices in $\mathcal{C} \setminus \mathcal{C}'$ and their subtrees from $T(v)$. Note that, for each j with $1 \le j \le k$, $T'(v_{i_j})$ still lies inside a box of height $|T(v_{i_j})|$ and width $rpw(T(v_{i_j}))$ on h. We define the *shrink subtree* procedure on T'_v as follows. We move each vertex v_{i_j} along with their subtrees from $\mathcal{C}_{T_v}(v_{i_j})$ to $(\mathcal{C}_{T_v}(v_{i_j})_x, \mathcal{C}_{T_v}(v_{i_j})_y, \mathcal{C}_{T'_v}(v_j)_z)$. Let us denote the shrunk subtree by $\mathcal{C'}_{T'_v}$. The height of the shrunk subtree $\mathcal{C'}_{T'_v}$ is equal to the number of vertices in $T'(v)$. Also, note that shrinking is a crossing-free unidirectional morph.

4 Morphing Through Lifting Paths

Let T be an n-vertex tree and $\mathcal{P}$ be a path decomposition of T into k paths. In this section, we describe an algorithm that morphs a plane drawing $\Gamma = \Gamma_0$ in XY_0 plane of tree T to the canonical 3D drawing $\Gamma' = \mathcal{C}(T)$ of T in $\mathcal{O}(k)$ steps. It lifts the paths of $\mathcal{P}$ one by one applying procedure *Lift*(). Note that the final positions for the vertices in $\mathcal{C}(T)$ are independent of $\mathcal{P}$. Also, a morph from $\mathcal{C}(T)$ to Γ' can be obtained by applying the morph from Γ' to $\mathcal{C}(T)$ backwards. At all times during the algorithm, the following invariant holds: a path $P_i \in \mathcal{P}$ is lifted only after all the children of the internal vertices of P_i are lifted. After the execution of *Lift*(P_i), path P_i moves to its canonical position with respect to *head*(P_i), see Fig. 2 and 3.

Step 0: Preprocessing. This step is a single stretching morph $\langle \Gamma, \Gamma_1 \rangle$ with $\mathcal{S}_1 = 2 \cdot (rpw + d(\Gamma))$. Note that stretching is a crossing-free morph.

***Lift*(*path*) Procedure**
Let $P_i = (v_0, v_1, \ldots, v_m)$ be the first path in $\mathcal{P}$ that has not been processed yet and Γ_t be the current drawing of T. We lift the path P_i. For any vertex v let *lifted subtree* $T'(v)$ be the portion of subtree $T(v)$ that has been lifted after execution of *Lift*(P_j) for some $j < i$. Let *the processing vertices* be the internal vertices of P_i along with the vertices of their lifted subtrees. The subtrees of all internal vertices v_j in P_i are already lifted due to the ordering among the paths. Suppose the lifted subtrees are in the canonical position with respect to the roots, the maximum height of vertices in an intermediate drawing Γ_t is strictly less than n and the difference of width between a lifted vertex and its root is at most rpw. We provide a brief overview of the *Lift*() procedure in the following.

The procedure *Lift*(P_i) consists of 13 steps and results in moving vertices of path P_i along with their lifted subtrees to their canonical positions with respect

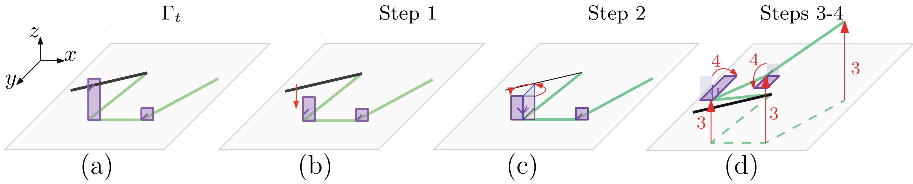

Fig. 2. (a) Drawing Γ_t in the beginning of the procedure *Lift*(P_i), bounding boxes for lifted subtrees are violet, P_i consists of green edges. Directions of movement of the vertices are shown with red arrows. (b) **Step 1**, (c) **Step 2** and (d) **Steps 3–4** of *Lift*(). (Color figure online)

to the head of P_i, i.e.,vertex v_0. Since the height of any lifted vertex is strictly less than n and the difference of width between a lifted vertex and its root is at most rpw, preprocessing Step 0 and Lemma 1 guarantee that the already lifted subtrees lie in the disjoint right circular cylinders of radius rpw and height n.

Step 1: For every internal vertex v_j of the path P_i, its lifted subtree $T'(v_j)$ morphs into the shrunk lifted subtree, see Sect. 3. All subtrees are shrunk simultaneously in one morphing step. This step is needed to ensure that the maximum height of a vertex does not exceed $2n$ during the *Lift*() procedure. It is a crossing-free morph since the subtrees move in mutually disjoint cylinders.

Step 2: It consists of steps $\langle \Gamma_t, \Gamma_{t+1}\rangle, \langle \Gamma_{t+1}, \Gamma_{t+2}\rangle$. For $0 \leq j < m-1$, if projection $pr(T'(v_j))$ overlaps with $pr((v_j, v_{j+1}))$, we rotate twice the drawing of $T'(v_j)$ around the vertical pole through $\Gamma_t(v_j)$. Since every lifted subtree $T'(v_j)$ lies in $XZ^+_{v_j}$, after this step all lifted subtrees lie in $XZ^+_{v_j}$ or $XZ^-_{v_j}$. It is a crossing-free morph since the rotations of subtrees happen inside mutually disjoint cylinders.

Step 3: In the morphing step $\langle \Gamma_{t+2}, \Gamma_{t+3}\rangle$, each internal vertex $v_j, j \geq 1$ of path P_i moves vertically to the height defined recursively as follows: for v_1: $\Gamma_{t+3}(v_1)_z = n$; for $v_j, j > 1$: $\Gamma_{t+3}(v_j)_z = \Gamma_{t+3}(v_{j-1})_z + |T'(v_{j-1})|$. Note that $|T'(v_j)|$, a number of vertices in $T'(v_j)$, is equal to the height of this shrinked lifted subtree. This step is crossing free since the projections of different subtrees and the path edges to the XY_0 plane does not change during the morph. After this step the vertices of P_i are in the same vertical order as in the canonical drawing $\mathcal{C}(T)$.

Step 4: The lifted subtree of each internal vertex of P_i is rotated to lie in a horizontal plane passing through the corresponding vertex. This step places all $T'(v_j)$ in disjoint horizontal planes. The direction of rotation is chosen in such a way that $T'(v_j)$ does not cross with an edge (v_j, v_{j+1}).

Steps 5 and 6: In Step 5 ($\langle \Gamma_{t+4}, \Gamma_{t+5}\rangle$), each vertex $v_j (j \geq 2)$ of the path P_i moves together with its subtree $T'(v_j)$ along the vector $((v_{1_x} - v_{j_x}) + \mathcal{C}(v_j)_x - \mathcal{C}(v_1)_x, v_{1_y} - v_{j_y}, 0)$, where v_{1_x} denotes x-coordinate of vertex v_1 in drawing Γ_{t+4}. In Step 6 ($\langle \Gamma_{t+5}, \Gamma_{t+6}\rangle$), every vertex $v_j, j \geq 2$ of the path P_i moves together with

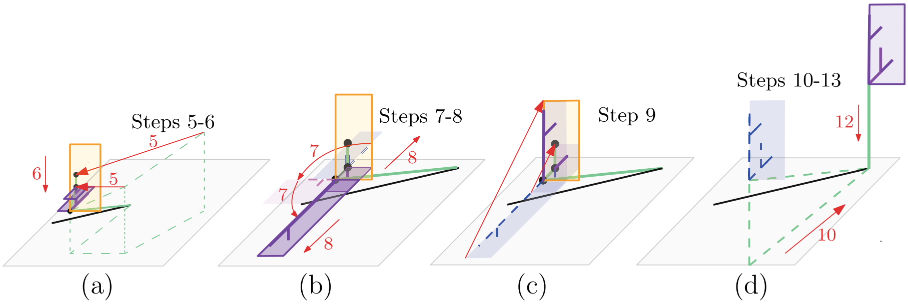

Fig. 3. Yellow plane is a vertical plane. (a) **Steps 5–6**, (b) **Steps 7–8**, (c) **Step 9** and (d) **Steps 10–13** (Steps 11, 13 do not make any changes in this example) of *Lift*(). (Color figure online)

its subtree $T'(v_j)$ along the same vertical vector $(0, 0, (v_{1_z} - v_{j_z}) + \mathcal{C}(v_j)_z - \mathcal{C}(v_1)_z)$, where v_{1_z} means z-coordinate of vertex v_1 in drawing Γ_{t+5}. Steps 5 and 6 move $v_2, \ldots, v_m$ to their canonical positions with respect to the vertex v_1

Steps 7, 8 and 9: Step 7, i.e., $\langle \Gamma_{t+6}, \Gamma_{t+7} \rangle, \langle \Gamma_{t+7}, \Gamma_{t+8} \rangle$, turns every lifted subtree $T'(v_j)$ of internal vertices of P_i to lie in positive x-direction with respect to v_j. Step 8, i.e., $\langle \Gamma_{t+8}, \Gamma_{t+9} \rangle$, morphs lifted subtrees of internal vertices of P_i in the horizontal planes from shrunk to the canonical size. In Step 9 ($\langle \Gamma_{t+9}, \Gamma_{t+10} \rangle$) all lifted subtrees $T'(v_j)$ of the internal vertices of P_i rotate around horizontal axes $(x, v_{j_y}, v_{j_z}), x \in \mathbb{R}$ to lie in vertical plane in positive direction such that the subtree $T(v_1)$ is in the canonical position with respect to v_1.

Step 10: In the morphing step $\langle \Gamma_{t+10}, \Gamma_{t+11} \rangle$, every internal vertex v_j of the path with its subtree $T'(v_j)$ moves horizontally in the direction $(v_{0_x} - v_{1_x}, v_{0_y} - v_{1_y}, 0)$. If in $\mathcal{C}(T)$ the edge (v_0, v_1) is vertical, vertex v_1 moves along this vector to get x, y-coordinates equal to (v_{0_x}, v_{0_y}). Otherwise, vertex v_1 moves along this vector as long as possible to get integer x and y coordinates not equal to (v_{0_x}, v_{0_y}). Step 10 ensures that Steps 11–13 move vertices only inside right circular cylinder of radius $rpw + d(\Gamma)$ and height $2n$ around v_0. During Steps 11–13 the processed part of the tree does not intersect with the unprocessed part since the above mentioned cylinders are disjoint for the vertices that are lying in XY_0.

Steps 11, 12 and 13: These steps differ depending on whether or not we have rotated $T'(v_0)$ during Step 2. In one case $T'(v_0)$ is in x-positive direction from v_0 and in the other $T(v_1)$ is in x-positive direction from v_0. Steps 11 and 13 make two rotations of the needed part of the tree to correct it's x, y-coordinates. Also, we need to move v_1 and its subtree to the canonical height with respect to v_0. In Step 12, we make the z-coordinate correction of $T(v_1)$. The steps are ordered in such a way that no intersections happen during their execution. Step 13 concludes the procedure *Lift*(P_i) by placing all processing vertices into their canonical positions with respect to v_0.

In the end of these morphing steps, we observe that all the internal vertices of P_i along with their subtrees are placed in the canonical position with respect to v_0. The lifted subtrees that were in the relative canonical position at the beginning of $Lift(P_i)$, still maintain their positions. For any path P_k, such that $k > i$, its vertices still lie on the XY_0 plane and their positions do not change during these steps. We keep on lifting up paths until we obtain the canonical drawing of T. The following theorem summarises what we achieved in this section.

Theorem 1. *For every two planar straight-line grid drawings Γ, Γ' of tree T with n vertices there exists a crossing-free 3D-morph $\mathcal{M} = \langle \Gamma = \Gamma_0, \ldots, \Gamma_l = \Gamma' \rangle$ that takes $\mathcal{O}(k)$ steps where k is number of paths in some path decomposition of tree T. In this morph, every intermediate drawing $\Gamma_i, 1 \leq i \leq l$ is a straight-line 3D grid drawing lying in a grid of size $\mathcal{O}(d^2 \times d^2 \times n)$, where d is maximum of the diameters of the given drawings.*

5 Morphing Through Lifting Edges

In this section, we describe another algorithm that morphs a planar drawing Γ of tree T to the canonical drawing $\mathcal{C}(T)$ of T. This time one iteration of our algorithm lifts simultaneously a set of edges with at most one edge of each path of a selected path decomposition. Let $\Gamma = \Gamma_0$ be a planar drawing of T.

Step 0: Preprocessing. This step $\langle \Gamma, \Gamma_1 \rangle$ is a stretching morph with $\mathcal{S}_1 = 2 \cdot rpw \cdot d(\Gamma) \cdot (4 \cdot d(\Gamma) + 1)$. It is a crossing-free morph.

$\overline{Lift}$(***edges***) ***procedure***
For edge e of T, let $st(e)$ (respectively, $end(e)$) be the vertex of e with smallest (respectively, largest) depth. Let $\mathcal{K} = \{K_1, \ldots, K_m\}$ be the partition of edges of T into disjoint sets such that $e \in K_i$ if and only if $dpt(st(e)) = m - i$, where m denotes the depth of T. We lift up sets K_i from $\mathcal{K}$ from $i = 1$ to $i = m$ by executing $\overline{Lift}(K_i)$ (Steps 1–5, see Fig. 4 and 5). Let Γ_t be the drawing of T before lifting set K_i. Let *lifted subtree* $T'(v_j)$ be the portion of subtree $T(v_j)$ lifted by the execution of $\overline{Lift}(K_j)$ where $j < i$. Suppose the drawing of $T'(v)$ in Γ_t is the canonical drawing of $T'(v)$ with respect to v; and the vertices that are incident to some non-processed edges lie in XY_0 plane.

Lemma 2. *For every edge $e = (v, u)$ with $st(e) = v$ in Γ_1 there is a lattice point $z_e \in e$ such that $B(\Gamma_1(z_e), rpw \cdot d(\Gamma)) \subset B(\Gamma_1(v), rpw \cdot d(\Gamma) \cdot (4 \cdot d(\Gamma) + 1))$. For distinct pair of edges $e_1, e_2 \in K_i \forall i = 1, \ldots, m$ disks $B(\Gamma_1(z_{e_1}), rpw)$ and $B(\Gamma_1(z_{e_2}), rpw)$ are disjoint. Also, for distinct pair of edges $e_1, e_2 \in K_i \forall i = 1, \ldots, m$ regions $\mathcal{F}_{e_1}, \mathcal{F}_{e_2}$ are disjoint, where $\mathcal{F}_e = \{x \in XY_0 : dist_{\Gamma_1}(x, (z_e, u)) \leq rpw\}$.*

Step 1: Shrink. In the step $\langle \Gamma_t, \Gamma_{t+1} \rangle$, for every edge $e \in K_i$ we move vertex $end(e)$ along with its lifted subtree towards $st(e)$ until $end(e)$ reaches point z_e.

Step 2: Go up. In morphing step $\langle \Gamma_{t+1}, \Gamma_{t+2} \rangle$, we move $end(e)$ with $T'(end(e))$ along the vector $(0, 0, \mathcal{C}(end(e))_z - \mathcal{C}(st(e))_z)$ for all $e \in K_i$.

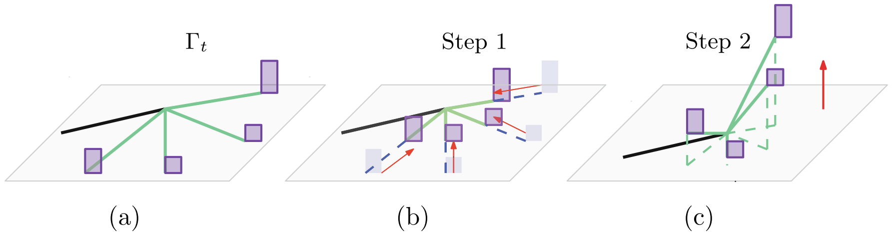

Fig. 4. (a) Drawing Γ_t in the beginning of the procedure $\overline{Lift}(K_i)$, bounding boxes for lifted subtrees are violet, K_i consists of green edges. (b) **Step 1** and (c) **Step 2** of $\overline{Lift}()$.

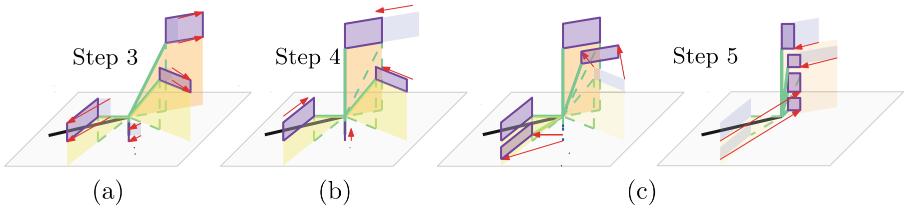

Fig. 5. (a) **Step 3**; (b) **Step 4**; (c) **Step 5**, consists of two morphing steps.

Step 3: Mapping. Morphing step $\langle \Gamma_{t+2}, \Gamma_{t+3} \rangle$ is a mapping morph, see Sect. 3. For every lifted subtree $T'(v_j)$, where $v_j = end(e), e \in K_i$, we define the half-planes of the mapping morph as follows: half-plane α is $XZ^+_{v_j}$, half-plane β is part of the vertical plane containing the edge e in such direction that $e \notin \beta$, the common vertical pole of α and β is a pole through v_j. All mapping steps are done simultaneously for all subtrees of end vertices of the edges of K_i.

Step 4: Shrink more. The morphing step $\langle \Gamma_{t+3}, \Gamma_{t+4} \rangle$ is a horizontal morph.

For each $v_j = end(e), e \in K_i$ we define a horizontal vector of movement as follows. If e is a vertical edges in canonical drawing then this vector is $(\Gamma_{t+3}(st(e))_x - \Gamma_{t+3}(end(e))_x, \Gamma_{t+3}(st(e))_y - \Gamma_{t+3}(end(e))_y, 0)$, in this case subtree $T'(end(e))$ is moving towards vertical pole through $st(e)$ until the image of the edge e becomes vertical. If e is not a vertical edge in canonical drawing, then $\mathcal{C}(end(e))_x - \mathcal{C}(st(e))_x = 1$ and we move the whole subtree $T'(end(e))$ towards the pole through $\Gamma_{t+3}(st(e))$ until $end(e)$ reaches the last point with integer coordinates before $(\Gamma_{t+3}(st(e))_x, \Gamma_{t+3}(st(e))_y, \Gamma_{t+3}(end(e))_z)$.

Step 5: Collide planes. During the following steps $\langle \Gamma_{t+4}, \Gamma_{t+5} \rangle, \ldots, \langle \Gamma_{t+5+\log k}, \Gamma_{t+5+\log k+1} \rangle$ we iteratively divide half-planes that contain $T'(end(e)), e \in K_i$ around each vertex $st(e), e \in K_i$ in pairs which are formed of neighboring half-planes in clockwise order around the pole through $st(e)$. If in some iteration there are odd number of planes around some pole, the plane without pair does not move in this iteration. In every iteration we map the drawing of one plane in the pair to another simultaneously in all pairs. As around each vertex we can have at most $k = \Delta(T)$ number of half-planes, we need at most $\mathcal{O}(\log k)$ number of mapping steps to collide all planes in one and to rotate the resulting image to $XZ^+_{st(e)}$

We perform $\overline{Lift}()$ for each $K_i \in \mathcal{K}$ till we obtain the canonical drawing of T. The following theorem summarises the result of this section.

Theorem 2. *For every two planar straight-line grid drawings Γ, Γ' of an n-vertex tree T, there exists a crossing-free 3D-morph $\mathcal{M} = \langle \Gamma = \Gamma_0, \dots, \Gamma_k = \Gamma' \rangle$ that takes $\mathcal{O}(dpt(T) \cdot \log \Delta(T))$ steps and $\mathcal{O}(d^3 \cdot \log n \times d^3 \cdot \log n \times n)$ space such that every intermediate drawing $\Gamma_i, 0 \le i \le k$ is a straight-line 3D grid drawing, where d is maximum of the diameters of the given drawings. In the worst case the algorithm can take $\mathcal{O}(dpt(T) \cdot \log n)$ steps since the maximum degree of T can be $\mathcal{O}(n)$.*

6 Trade-off

Recall that $\mathcal{L}(T)$ is the set of paths induced by the long-path decomposition, see Sect. 2. Let $Long(T)$ be a set of paths from $\mathcal{L}(T)$, consisting of the paths whose length is at least $\sqrt{n}$, i.e. $Long(T) = \{L_i \in \mathcal{L}(T) : |L_i| \ge \sqrt{n}\}$, let the order in $Long(T)$ be induced from the order in $\mathcal{L}(T)$. We denote by $Short(T)$ a set of trees that are left after deleting from T edges of $Long(T)$.

Lemma 3. *$|Long(T)| \le \sqrt{n}$ and for every tree T_i in $Short(T)$ depth of T_i is at most $\lfloor \sqrt{n} \rfloor$.*

We divide edges in $Short(T)$ into disjoint sets $Sh_1, \dots Sh_{\lfloor \sqrt{n} \rfloor}$. An edge (v_i, v_j) in tree T_k lies in the set Sh_l if and only if $max(dpt(v_i), dpt(v_j)) = \lfloor \sqrt{n} \rfloor - l + 1$, where $dpt(v)$ is the depth of vertex v in the corresponding tree T_k. Since the maximum depth of any tree T_k is at most $\sqrt{n}$, $Sh_1, \dots Sh_{\lfloor \sqrt{n} \rfloor}$ contain all the edges of these subtrees.

***Trade-off Algorithm*:** In the beginning we perform a stretching step with $\mathcal{S}_1 = 2 \cdot rpw \cdot d(\Gamma) \cdot (4 \cdot d(\Gamma) + 1)$ as mentioned in Sect. 5. $\mathcal{S}_1$ is big enough to perform *Lift*() procedure mentioned in Sect. 4. Then, we lift edges from sets Sh_1 to $Sh_{\lfloor \sqrt{n} \rfloor}$ by $\overline{Lift}(Sh_i)$ procedure. It takes $\mathcal{O}(\sqrt{n} \cdot \log \Delta(T))$ steps in total by Theorem 2. After that, we lift paths in $Long(T)$ in the order induced by the path decomposition. As $|Long(T)| \le \sqrt{n}$ and each *Lift*() procedure consists of a constant number of morphing steps, this step takes $\mathcal{O}(\sqrt{n})$ steps.

Theorem 3. *For every two planar straight-line grid drawings Γ, Γ' of tree T with n vertices there exists a crossing-free 3D-morph $\mathcal{M} = \langle \Gamma = \Gamma_0, \dots, \Gamma_l = \Gamma' \rangle$ that takes $\mathcal{O}(\sqrt{n} \cdot \log \Delta(T))$ steps ($\mathcal{O}(\sqrt{n} \cdot \log n)$ in the worst case) and $\mathcal{O}(d^3 \cdot \log n \times d^3 \cdot \log n \times n)$ space to perform, where d is maximum of the diameters of the given drawings. In this morph every intermediate drawing $\Gamma_i, 1 \le i \le l$ is a straight-line 3D grid drawing. It is possible to morph between Γ, Γ' using $\mathcal{O}(\sqrt{n})$ steps if maximum degree of T is a constant.*

7 Conclusion

In this paper, we presented an algorithm that morphs between two planar grid drawings of an n-vertex tree T in $\mathcal{O}(\sqrt{n}\log n)$ steps such that all intermediate drawings are crossing-free $3D$ grid drawings and lie inside a polynomially bounded $3D$-grid. Arseneva et al. [2] proved that $\mathcal{O}(\log n)$ steps are enough to morph between two planar grid drawings of an n-vertex tree T where intermediate drawings are allowed to lie in $\mathbb{R}^3$ but they did not guarantee that intermediate drawings have polynomially bounded resolution. Several problems are left open in this area of research. We mention some of them here. It is interesting to prove a lower bound on the number of morphing steps if intermediate drawings are allowed to lie in $\mathbb{R}^3$ (with or without the additional constraint of polynomially bounded resolution). Another intriguing question is if it possible to morph between two planar grid drawings in $o(n)$ number of steps for a richer class of graphs (e.g. outer-planar graphs) than trees if we are allowed to use the third dimension.

Acknowledgements. Elena Arseneva was partially supported by the Foundation for the Advancement of Theoretical Physics and Mathematics "BASIS". Elena Arseneva and Aleksandra Istomina were partially supported by RFBR, project 20-01-00488. Rahul Gangopadhyay was supported by Ministry of Science and Higher Education of the Russian Federation, agreement no. 075-15-2019-1619.

References

1. Alamdari, S., et al.: How to morph planar graph drawings. SIAM J. Comput. **46**(2), 824–852 (2017). https://doi.org/10.113716M1069171
2. Arseneva, E., et al.: Pole dancing: 3D morphs for tree drawings. J. Graph Algorithms Appl. **23**(3), 579–602 (2019)
3. Barrera-Cruz, F., et al.: How to morph a tree on a small grid. In: Friggstad, Z., Sack, J.-R., Salavatipour, M.R. (eds.) WADS 2019. LNCS, vol. 11646, pp. 57–70. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24766-9_5
4. Bender, M.A., Farach-Colton, M.: The level ancestor problem simplified. In: Rajsbaum, S. (ed.) LATIN 2002. LNCS, vol. 2286, pp. 508–515. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45995-2_44
5. Biedl, T.: Optimum-width upward drawings of trees. arXiv preprint arXiv:1506.02096 (2015)