SIAM J. COMPUT.
Vol. 46, No. 2, pp. 824–852

HOW TO MORPH PLANAR GRAPH DRAWINGS[*]

SOROUSH ALAMDARI[†], PATRIZIO ANGELINI[‡], FIDEL BARRERA-CRUZ[§], TIMOTHY M. CHAN[¶], GIORDANO DA LOZZO[‖], GIUSEPPE DI BATTISTA[#], FABRIZIO FRATI[#], PENNY HAXELL[††], ANNA LUBIW[¶], MAURIZIO PATRIGNANI[#], VINCENZO ROSELLI[#], SAHIL SINGLA[‡‡], AND BRYAN T. WILKINSON[§§]

Abstract. Given an n-vertex graph and two straight-line planar drawings of the graph that have the same faces and the same outer face, we show that there is a morph (i.e., a continuous transformation) between the two drawings that preserves straight-line planarity and consists of $O(n)$ steps, which we prove is optimal in the worst case. Each step is a *unidirectional linear morph*, which means that every vertex moves at constant speed along a straight line, and the lines are parallel although the vertex speeds may differ. Thus we provide an efficient version of Cairns' 1944 proof of the existence of straight-line planarity-preserving morphs for triangulated graphs, which required an exponential number of steps.

Key words. planar graphs, transformation, morph

AMS subject classifications. 68R10, 05C10

DOI. 10.1137/16M1069171

1. Introduction.

A morph between two geometric shapes is a continuous transformation of one shape into the other. Morphs are useful in many areas of computer science—computer graphics, animation, and modeling, to name just a few. The usual goal in morphing is to ensure that the structure of the shapes be "visible" throughout the entire transformation.

[*] Received by the editors April 5, 2016; accepted for publication (in revised form) December 28, 2016; published electronically April 27, 2017.

http://www.siam.org/journals/sicomp/46-2/M106917.html

Funding: The second author was partially supported by DFG grant Ka812/17-1. The fourth and ninth authors were partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. The fifth author was partially supported by the U.S. Defense Advanced Research Projects Agency (DARPA) under agreement AFRL FA8750-15-2-0092. The sixth, seventh, tenth, and eleventh authors were partially supported by H2020-MSCA-RISE project 73499—"CONNECT," by MIUR project "AMANDA – Algorithmics for MAssive and Networked DAta," 2012C4E3KT_001, and by MIUR project "MODE – MOrphing graph Drawings Efficiently," 20157EFM5C_001. The eighth author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. The thirteenth author was supported by the Danish National Research Foundation grant DNRF84 through the Center for Massive Data Algorithmics (MADALGO).

[†] Department of Computer Science, Cornell University, Ithaca, NY 14853 (alamdari@cs.cornell.edu).

[‡] Wilhelm-Schickard-Institut für Informatik, Tübingen University, Germany 72076 (angelini@informatik.uni-tuebingen.de).

[§] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (fidelbc@math.gatech.edu).

[¶] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (tmchan@uwaterloo.ca, alubiw@uwaterloo.ca).

[‖] Department of Computer Science, University of California, Irvine, CA (gdalozzo@uci.edu). The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

[#] Department of Engineering, Roma Tre University, Rome, 00146, Italy (gdb@dia.uniroma3.it, frati@dia.uniroma3.it, patrigna@dia.uniroma3.it, roselli@dia.uniroma3.it).

[††] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (pehaxell@uwaterloo.ca).

[‡‡] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (ssingla@cmu.edu).

[§§] Center for Massive Data Algorithmics, Aarhus University, Aarhus, Denmark (wilkinson@cs.au.dk).

Two-dimensional graph drawings can be used to represent many of the shapes for which morphs are of interest, e.g., two-dimensional images [11, 24, 41], polygons, and polylines [1, 3, 14, 21, 26, 32, 35, 36, 37]. For this reason, morphs of graph drawings have been well studied.

For the problem of morphing graph drawings, the input consists of two drawings Γ_0 and Γ_1 of the same graph G, and the problem is to transform continuously from the first drawing to the second drawing. A *morph* between Γ_0 and Γ_1 is a continuously changing family of drawings of G indexed by time $t \in [0, 1]$, such that the drawing at time $t = 0$ is Γ_0 and the drawing at time $t = 1$ is Γ_1. Maintaining structure during the morph becomes a matter of preserving geometric properties such as planarity, straight-line planarity, edge lengths, or edge directions. For example, preserving edge lengths in a straight-line drawing leads to problems of linkage reconfiguration [16, 18].

In this paper we consider the problem of morphing between two graph drawings while preserving planarity. Of necessity, we assume that the initial and final planar drawings are *topologically equivalent*—i.e., have the same faces and the same outer face. In addition to the above-mentioned applications, morphing graph drawings while preserving planarity has application to the problem of creating three-dimensional models from two-dimensional slices [8], with time playing the role of the third dimension.

When planar graphs may be drawn with polyline edges the morphing problem becomes much easier—the intuition is that vertices can move around while edges bend to avoid collisions. An efficient morphing algorithm for this case was given by Lubiw and Petrick [30]. The case of orthogonal graph drawings is also well-solved—Biedl et al. [12] gave an algorithm to morph efficiently between any two orthogonal drawings of the same graph while preserving planarity and orthogonality.

We restrict our attention in this paper to straight-line planar drawings. Our main result is an efficient algorithm to morph between two topologically equivalent straight-line planar drawings of a graph, where the morph must preserve straight-line planarity. The issue is to find the vertex trajectories, since the edges are determined by the vertex positions.

Existence of such a morph is not obvious, and was first proved in 1944 by Cairns [13] for the case of triangulations. Cairns used an inductive proof, based on contracting a low-degree vertex to a neighbor. In general, a contraction that preserves planarity in both drawings may not exist, so Cairns needed a preliminary morphing procedure to make this possible. As a result, his method involved two recursive calls, and took an exponential number of steps. Thomassen [42] extended the proof to all planar straight-line drawings. He did this by augmenting both drawings to isomorphic ("compatible") triangulations which reduces the general case to Cairns's result. The idea of compatible triangulations was rediscovered and thoroughly explored by Aronov, Seidel, and Souvaine [7], who showed, among other things, that two drawings of a graph on n vertices have a compatible triangulation of size $O(n^2)$ and that this bound is tight in the worst case.

Floater and Gotsman [22] gave an alternative way to morph straight-line planar triangulations based on Tutte's graph drawing algorithm [44]. Gotsman and Surazhsky [25, 38, 39, 40] extended the method to all straight-line planar graph drawings using the same idea of compatible triangulations, and they showed that the resulting morphs are visually appealing. These algorithms do not produce explicit vertex trajectories; instead, they compute the intermediate drawing (a "snapshot") at any requested time point. There are no quality guarantees about the of time points required to approximate continuous motion while preserving planarity. For related results, see [19, 20, 23].

For more history and related results on morphing graph drawings, see Roselli's PhD thesis [34, section 3.1].

The problem of finding a straight-line planarity-preserving morph that uses a polynomial number of discrete steps has been investigated several times (see, e.g., [28, 29, 30, 31]). The most natural definition of a discrete step is a *linear morph*, where every vertex moves along a straight-line segment at uniform speed. Note that we do not require that all vertices move at the same speed. One of the surprising things we discovered (after our first conference version [2]) is that it is actually easier to solve our problem using a more restrictive type of linear morph. Specifically, we define a morph to be *unidirectional* if every vertex moves along a straight-line segment at uniform speed, and all the lines of motion are parallel. As a special case, a linear morph that only moves one vertex is unidirectional by default.

1.1. Main result.

The main result of this paper is the following.

THEOREM 1.1. *Given a planar graph G on n vertices and two straight-line planar drawings of G with the same faces and the same outer face, including the same nesting of connected components, there is a morph between the two drawings that preserves straight-line planarity and consists of $O(n)$ unidirectional morphs. Furthermore, the morph can be found in time $O(n^3)$.*

This paper combines four conference papers: [2] designs a general algorithmic scheme for constructing morphs between planar graph drawings and proves the first polynomial upper bound on the number of morphing steps; [9] introduces unidirectional morphs; [6] introduces techniques to handle nontriangulated graphs; [4] solves a crucial subproblem of "convexifying a quadrilateral" with a single unidirectional morph, yielding a linear bound on the total number of morphing steps (which is proved optimal). New to this version is the handling of disconnected graphs.

Techniques from our paper have been used in algorithms to morph Schnyder drawings [10], and algorithms to morph convex drawings while preserving convexity [5].

From a high-level perspective, our proof of Theorem 1.1 has two parts: to solve the problem for the special case of a maximal planar graph, in which case both drawings are triangulations, and to reduce the general problem to this special case.

Previous papers [2, 25, 42] reduced the general case to the case of triangulations by finding "compatible" triangulations of both drawings, which increases the size of the graph to $O(n^2)$. We improve this by making use of the freedom to morph the drawings. Specifically, we show that after a sequence of $O(n)$ unidirectional morphs we can triangulate both drawings with the same edges. Thus we reduce the general problem to the case of triangulations with the same input size.

For the case of triangulations, the main idea of our algorithm is the same one that Cairns [13] used to prove existence of a morph, namely, to find a vertex v that can be contracted in both drawings to a neighbor u while preserving planarity. Contracting v to u gives us two planar drawings of a smaller graph. Using recursion, we can find a morph between the smaller drawings that consists of unidirectional morphs. Thus the total process, which is illustrated in Figure 1, is to move v along a straight line to u in the first drawing (a unidirectional morph since only one vertex moves), perform the recursively computed morph, and then, in the second drawing, reverse the motion of v to u.

Note that this process allows a vertex to become coincident with another vertex, so it is not a true morph, but rather what we call a *pseudomorph*. There are two

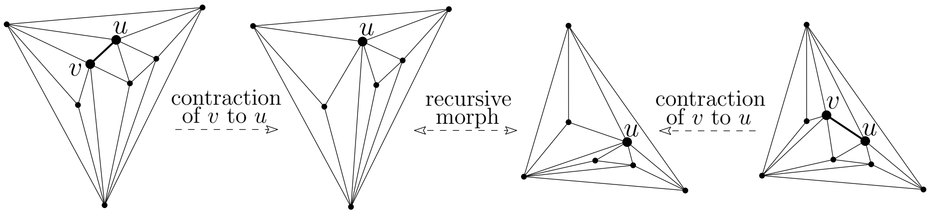

FIG. 1. *A schema for a morphing algorithm for triangulations.*

main issues with this plan: (1) to deal with the fact that there may not be a vertex v that can be contracted to the same neighbor u in both drawings, and (2) to convert a pseudomorph to a morph. We will give a few more details on each of these.

A contractible vertex always exists if we are dealing with a single triangulation: because the graph is planar, there is an internal vertex v of degree less than or equal to 5; because the graph is triangulated, the neighbors of v form a polygon of at most 5 vertices; and by an easy geometric argument, such a polygon always has a vertex u that "sees" the whole polygon, so v can be contracted to u while preserving planarity. In our situation there is one complication—we want to contract v to the same neighbor in both drawings. This is easy to solve in the case where our low-degree vertex v has neighbors that form a convex polygon in the first drawing—in this case every neighbor of v sees the whole convex polygon in the first drawing, so we can choose the vertex u that works in the second drawing. Our general solution will be to morph the first drawing so that v's neighbors form a convex polygon (or at least "convex enough", in the sense that u sees the whole polygon). This was the same approach that Cairns used, though his solution required an exponential number of morphing steps, and our solution will only require two morphing steps.

Summarizing the first part of our algorithm to morph between two triangulations, we show that after two unidirectional morphs we can obtain a vertex v that can be contracted to the same neighbor in both drawings.

After performing the contraction we apply induction to find a morph (composed of unidirectional morphs) between the two smaller drawings. The last issue is to convert the resulting pseudomorph to a true morph. Instead of contracting v to a neighbor u, we must keep v close to, but not coincident with, u, while we follow the morph of the smaller graph. Cairns solved this issue by keeping v at the centroid of its surrounding polygon, but this results in nonlinear motion for v [2]. We will find a position for v in each drawing during the course of the morph so that the linear motion from one drawing to the next remains unidirectional.

Putting together the two parts of our algorithm, the total number of unidirectional morphs satisfies $S(n) = S(n-1) + O(1)$, which solves to $O(n)$.

This completes the high-level overview of our algorithm. From a low-level perspective, the heart of our algorithm is a solution to a problem we call *Quadrilateral Convexification*: given a triangulation containing a nonconvex quadrilateral, morph the triangulation to make the quadrilateral convex.

Our morphing algorithm uses $O(n)$ calls to Quadrilateral Convexification, plus $O(n)$ other unidirectional morphs. We will show that Quadrilateral Convexification can be accomplished with a single unidirectional morph, and thus our total bound is $O(n)$ unidirectional morphs. Our solution to the Quadrilateral Convexification problem is achieved via a connection to the existence of hierarchical planar convex drawings of hierarchical triconnected planar graphs.

We also show that the linear bound of Theorem 1.1 is asymptotically optimal in the worst case, namely, we have the following.

THEOREM 1.2. *There exist two straight-line planar drawings of an n-vertex path such that any straight-line planarity preserving morph between them that consists of k linear morphs is such that $k \in \Omega(n)$.*

In particular, we show that morphing from an n-vertex spiral to a straight path takes $\Omega(n)$ linear morphs by defining a measure of the difference between the two drawings that begins at $\Omega(n)$ and changes only by $O(1)$ during a single linear morph.

We remark that the lower bound of Theorem 1.2 does not assume the single morphing steps to be necessarily unidirectional.

The rest of the paper is organized as follows. First, in section 2 we give formal definitions of our terms and concepts. Then in section 3 we give a more detailed outline of the algorithm. We fill in solutions to the various subproblems in sections 4–7. In section 8 we present our lower bound. Finally, in section 9 we conclude and present some open problems.

2. Definitions and preliminaries.

A *straight-line planar drawing* Γ of a graph $G(V, E)$ maps vertices in V to distinct points of the plane and edges in E to non-intersecting open straight-line segments between their end vertices. A planar drawing of a graph partitions the plane into topological connected regions called *faces*. The unbounded face is called the *outer face*. Two planar drawings of a connected planar graph are *topologically equivalent* if they induce the same circular ordering of the edges around each vertex and have the same outer face. Two planar drawings of a disconnected planar graph are *topologically equivalent* if each connected component is topologically equivalent in both drawings and, furthermore, the connected components are nested the same way in both drawings. A *planar embedding* is an equivalence class of planar drawings of the same graph. A *plane* graph is a planar graph with a given planar embedding.

Given a vertex v of a graph G, the *neighbors* of v are the vertices adjacent to v, and the *degree* of v in G, denoted by $\deg(v)$, is the number of neighbors of v.

In a plane graph, a *facial cycle* is a closed walk that progresses from one edge xy to the next edge yz in the clockwise cyclic order of edges around vertex y. Note that a facial cycle is not necessarily simple if the graph is not biconnected. In a planar drawing, each inner face is bounded by an outer facial cycle and some number of inner facial cycles.

Morphs. If Γ_0 and Γ_1 are two drawings of the same graph, a *morph* between Γ_0 and Γ_1 is a continuously changing family of drawings of G indexed by time $t \in [0, 1]$, such that the drawing at time $t = 0$ is Γ_0 and the drawing at time $t = 1$ is Γ_1. In this paper we are only concerned with graph drawings in which every edge is drawn as a straight-line segment. In this case, a morph is specified by the vertex trajectories.

A *linear morph* is a morph in which every vertex moves along a straight-line segment at uniform speed. A linear morph is completely specified by the initial and final vertex positions. If vertex v is at position v_0 in the initial drawing (at time $t = 0$) and at position v_1 in the final drawing (at time $t = 1$), then its position at time t during a linear morph is $(1 - t)v_0 + tv_1$ for any $0 \leq t \leq 1$. Note that vertices may move at different speeds and, in particular, some vertices may remain stationary.

If Γ_0 and Γ_1 are straight-line planar drawings of a graph, we use $\langle \Gamma_0, \Gamma_1 \rangle$ to denote the linear morph from Γ_0 to Γ_1. We seek a morph that consists of a sequence of k linear morphs. Such a morph can be specified by $k + 1$ planar straight-line graph drawings.

If $\Gamma_1, \dots, \Gamma_{k+1}$ are straight-line planar drawings of a graph, we use $\langle \Gamma_1, \dots, \Gamma_{k+1} \rangle$ to denote the morph from Γ_1 to Γ_{k+1} that consists of the sequence of k linear morphs $\langle \Gamma_i, \Gamma_{i+1} \rangle$ for $i = 1, \dots, k$.

A *unidirectional morph* is a linear morph in which every vertex moves parallel to the same line, i.e., there is a line L with unit direction vector $\bar{\ell}$ such that each vertex moves linearly from an initial position v_0 to a final position $v_0 + k_v \bar{\ell}$ for some $k_v \in \mathbb{R}$. Note that k_v may be positive or negative and that different vertices may move different amounts along direction $\bar{\ell}$. We call this an *L-directional morph.* Observe that a linear morph of a single vertex is by default a unidirectional morph.

In this paper we restrict attention to topologically equivalent straight-line planar drawings and to morphs in which every intermediate drawing is straight-line planar. From now on we use the term *morph* to mean a *straight-line planarity preserving morph.* In particular, during the course of the morph, a vertex may not become coincident with another vertex, nor hit a nonincident edge.

In several places we will make use of the following basic result: If we have a straight-line planar-drawing whose outer face is a triangle, and if we apply a unidirectional morph to the three vertices of the triangular outer face, preserving the orientation of the triangle, and let the interior vertices follow along linearly, then the result is a unidirectional morph.

LEMMA 2.1. *Let x, y, z be the clockwise-ordered vertices of the triangular outer face of a straight-line planar drawing. Suppose that vertices x, y, and z move linearly in the direction of a vector $\bar{\ell}$ in such a way that their clockwise order is preserved. Any point p inside the triangle can be defined as a convex combination of x, y, and z, and in this way the motion of x, y, and z determines the motion of p. The result is a unidirectional morph of the straight-line planar drawing (in particular, planarity is preserved).*

Proof. We will prove the result for a maximal planar graph. This suffices, since any straight-line planar drawing can be augmented to a triangulation. (Observe that we do not need to compute such a triangulation when applying the lemma.)

Suppose that point p is defined by the convex combination $\lambda_1 x + \lambda_2 y + \lambda_3 z$ where $\sum \lambda_i = 1$ and $\lambda_i \geq 0$. Suppose the morph is indexed by $t \in [0, 1]$ and that the positions of the vertices at time t are x_t, y_t, z_t, p_t. Suppose that x moves by $k_1 \bar{\ell}$, y moves by $k_2 \bar{\ell}$, and z moves by $k_3 \bar{\ell}$. Thus $x_t = x_0 + t k_1 \bar{\ell}$, etc. Then

$$p_t = \lambda_1 x_t + \lambda_2 y_t + \lambda_3 z_t = \lambda_1 x_0 + \lambda_2 y_0 + \lambda_3 z_0 + t(\lambda_1 k_1 + \lambda_2 k_2 + \lambda_3 k_3)\bar{\ell} = p_0 + t k \bar{\ell},$$

where $k = \lambda_1 k_1 + \lambda_2 k_2 + \lambda_3 k_3$. Thus p also moves linearly in direction $\bar{\ell}$.

The fact that planarity is preserved follows from far more general results: The transformation of points x, y, z determines an affine transformation of the plane, and by hypothesis, this affine transformation preserves the orientation of triangle xyz. Affine transformations preserve convex combinations—thus our definition of the movement of any interior point p is the same as applying the affine transformation to p. An affine transformation that preserves the orientation of one triangle preserves the orientations of all triangles. This implies that the drawing is planar at all time points of the morph. $\square$

Geometry and triangulations. We will assume that our input graph drawings have vertices in general position, that is, no three vertices lie on the same line. We can achieve this property by a linear number of preliminary unidirectional morphing steps that slightly perturb the positions of the vertices.

A *triangulation* is a straight-line planar drawing of a maximal planar graph. Every face in a triangulation (including the outer face) is a triangle. The three vertices of the outer face are called *boundary vertices* and the others are called *internal* vertices.

If v is an internal vertex of a triangulation, we use $\Delta(v)$ to denote the polygon formed by the neighbors of v. For a simple polygon in the plane, the *kernel* of the polygon consists of the points inside the polygon from which the whole polygon is visible. Note that the kernel of any polygon is convex. The following result was noted by Cairns [13] and can be proved by simple case analysis.

LEMMA 2.2. *If P is a polygon with four or five vertices (a quadrilateral or a pentagon) then at least one vertex of P is contained in the kernel of P.*

We will use the result in the following form: if v is an internal vertex of degree at most 5 in a triangulation, then $\Delta(v)$ has a vertex in its kernel.

Contractions and pseudomorphs. A main tool we use in our morphing algorithm is vertex contraction in a triangulation. Contracting edge uv in a graph has the standard meaning, namely, we replace u and v by a new vertex adjacent to all the neighbors of u and v. We now define contraction in a triangulation. Let Γ be a drawing of a maximal planar graph G. Let v be an internal vertex and let u be a neighbor of v that lies in the kernel of $\Delta(v)$. *Contracting* v to u means moving v linearly from its original position to u while all other vertices remain fixed. Because the kernel is convex, every intermediate drawing is straight-line planar. Thus, this is a morph (and, in fact, a unidirectional morph) except for the final drawing in which v becomes coincident with u. By our general position assumption the final drawing is a straight-line planar drawing of the graph formed by contracting edge uv.

Our algorithm for morphing between two triangulations Γ_1 and Γ_2 works by contracting some vertex v to the same neighbor u in both drawings, and then recursively morphing between the two smaller triangulations. Expressing this as a transformation from Γ_1 to Γ_2, we contract v to u in Γ_1, apply the recursively computed morph, and then reverse the contraction of v to u in Γ_2. We call this last step "uncontraction" of v. We define a *pseudomorph* of a triangulation recursively in the above narrow sense: a *pseudomorph* consists either of (i) a unidirectional morphing step followed by a pseudomorph, (ii) a pseudomorph followed by a unidirectional morphing step, or (iii) a contraction of some vertex v to u, followed by a pseudomorph of the resulting smaller triangulation, followed by the uncontraction of v. A pseudomorph is not a morph because v becomes coincident with u. One of our main technical contributions is to show that every pseudomorph that is composed of unidirectional morphs can be converted to a "true" morph with the same number of unidirectional steps.

Our model of computation is a real random access machine (RAM), a standard model of computation used in computational geometry [33].

3. Overview of the algorithm.

In this section we describe all the ingredients of our algorithm and how they fit together.

Our most basic building block is an algorithm to morph a triangulation so that a given quadrilateral formed by two adjacent triangles becomes convex. One chord of the quadrilateral will lie inside the quadrilateral. A necessary condition is that the other chord not be part of the triangulation. Specifically, we solve the following problem (see Figure 2).

PROBLEM 3.1 (Quadrilateral Convexification). *Given an n-vertex triangulation Γ and given a quadrilateral $abcd$ in Γ with no vertex inside it and such that neither*

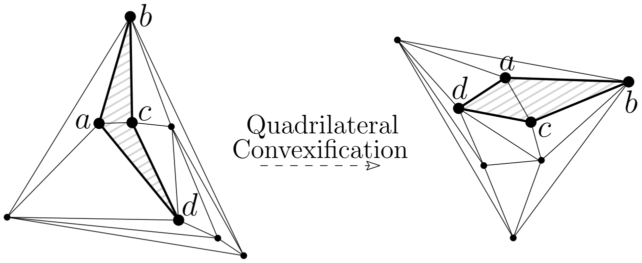

FIG. 2. *The quadrilateral convexification problem. Figure* 11 *illustrates our solution to the quadrilateral convexification problem, consisting of a single unidirectional morph.*

ac nor bd is an edge outside of abcd (i.e., abcd does not have external chords), morph Γ *so that abcd becomes convex.*

We solve Quadrilateral Convexification in section 6 giving the following result:

THEOREM 3.2. *Quadrilateral Convexification can be solved via a single unidirectional morph. Furthermore, such a morph can be found in* $O(n^2)$ *time.*

We will prove our main result, Theorem 1.1, by finding a morph that consists of $O(n)$ calls to Quadrilateral Convexification plus $O(n)$ further unidirectional morphs. Together with the above Theorem 3.2, this gives a total bound of $O(n)$ unidirectional morphs.

The proof of Theorem 1.1 has two parts. In section 4 we reduce the problem to the case of triangulations. Specifically, we show that given two topologically equivalent straight-line planar drawings of a graph G on n vertices, we can enclose each drawing with a triangle and then morph and add (the same) edges to triangulate both drawings, using $O(n)$ unidirectional morphs; this results in two triangulations which are topologically equivalent drawings of the same maximal planar graph. Note that a morph between these augmented drawings provides a morph of the originals by ignoring the added edges.

In section 5 we prove Theorem 1.1 for the case of triangulations, using $O(n)$ unidirectional morphs. Thus the two sections together prove Theorem 1.1.

In all cases when we say that we find a morph, we actually find a pseudomorph (as defined in the previous section) and rely on the following theorem (which is proved in section 7) to convert the pseudomorph to a true morph.

THEOREM 3.3. *Let* Γ_1 *and* Γ_2 *be two triangulations that are topologically equivalent drawings of an n-vertex maximal planar graph* G*. Suppose that there is a pseudomorph from* Γ_1 *to* Γ_2 *in which we contract an internal vertex* v *of degree at most* 5*, perform* k *unidirectional morphs, and then uncontract* v*. Then there is a morph* $\mathcal{M}$ *from* Γ_1 *to* Γ_2 *that consists of* $k+2$ *unidirectional morphs. Furthermore, given the sequence of* $k+1$ *drawings that define the pseudomorph, we can modify them to obtain the sequence of drawings that define* $\mathcal{M}$ *in* $O(k+n)$ *time.*

Note that although the proofs of Theorems 3.2 and 3.3 are deferred until the last two sections of the paper, they come first in terms of the dependency of results.

4. Morphing to find a compatible triangulation. In this section we show how to morph two topologically equivalent straight-line planar drawings of a graph G so that after the morph both drawings can be triangulated by adding the same edges. We allow G to be disconnected.

Theorem 4.1. *Let G be a planar graph with n vertices and c connected components. Given two topologically equivalent straight-line planar drawings of G, we can enclose each drawing in a triangle $z_1z_2z_3$ and then morph and add edges to create two triangulations that are topologically equivalent drawings of a maximal planar graph on vertex set $V(G) \cup \{z_1, z_2, z_3\}$. The morph consists of $O(c)$ unidirectional morphs plus $O(n)$ calls to Quadrilateral Convexification, for a total bound of $O(n)$ unidirectional morphs. The algorithm takes time $O(n^3)$.*

Proof. Let Γ_1 and Γ_2 be the two topologically equivalent drawings of G. We begin by adding a large triangle $z_1z_2z_3$ that encloses each drawing.

Our algorithm has two parts. In Part A we morph and add edges within connected components so that in any connected component of three or more vertices, each facial walk is a triangle. In Part B we add edges to connect the disconnected components and complete the triangulation.

Part A. Suppose we have a facial walk that has four or more vertices. We will find two consecutive edges uv and vw of the facial walk so that edge uw can be added to the graph, i.e., such that $u \neq w$ and uw is not an edge of the graph. Suppose first that the facial walk has two consecutive edges x_1x_2 and x_2x_3 that belong to different biconnected components of the graph, i.e., x_2 is a cut vertex whose removal disconnects the connected component x_2 belongs to. Then $x_1 \neq x_3$ and edge x_1x_3 does not belong to the graph, hence it can be added inside the face. If two such edges x_1x_2 and x_2x_3 do not exist, then the facial walk is a simple cycle. Then consider four consecutive vertices x_1, x_2, x_3, and x_4 along the cycle. By planarity, edge x_1x_3 or edge x_2x_4 does not belong to the graph, hence it can be added inside the face.

At this point we have two consecutive edges uv and vw of the facial walk so that edge uw can be added to the graph while preserving the simplicity of the graph, as in Figure 3(a). We will morph drawings Γ_1 and Γ_2 so that edge uw can be added as a straight-line segment preserving planarity. (In case our facial walk bounds an inner face that contains no disconnected components, we could have chosen uw to be an ear of a triangulation of the face in one of the drawings, thus avoiding the need to morph that drawing; but in the general case we must morph both drawings.) The argument is the same for both drawings, so we describe it only for Γ_1. Vertices u and w are consecutive neighbors of v. Add a new neighbor r of v between u and w in cyclic order, and add edges rv, ru, and rw. It is possible to place r close enough to v in Γ_1 so that the resulting drawing is straight-line planar and (v, r, u) and (v, r, w) are faces. See Figure 3(b) for an example.

We will now morph the resulting drawing to make the quadrilateral $urwv$ convex, as in Figure 3(c). To do this, we temporarily triangulate the drawing of the entire graph[1] and then apply quadrilateral convexification to $urwv$.

There is one slight complication: it might happen that when we triangulate the drawing, we add the edge uw, which would make it impossible to convexify the quadrilateral $urwv$. In this case, we remove uw and retriangulate the resulting quadrilateral by adding a new vertex p and adding straight-line edges from p to the four vertices of the quadrilateral. In particular, if edge uw was an internal edge of the triangulation, then p can be placed at any internal point of the line segment uw. Otherwise, let $uxwy$ be the clockwise order of the vertices incident to the outer face after the removal of edge uw; then p can be placed at any point to the left of the half-line from x through w and to the right of the half-line from y through w. See Figure 4.

[1] Any (possibly disconnected) polygonal domain with n vertices can be triangulated in time $O(n \log n)$ [17].

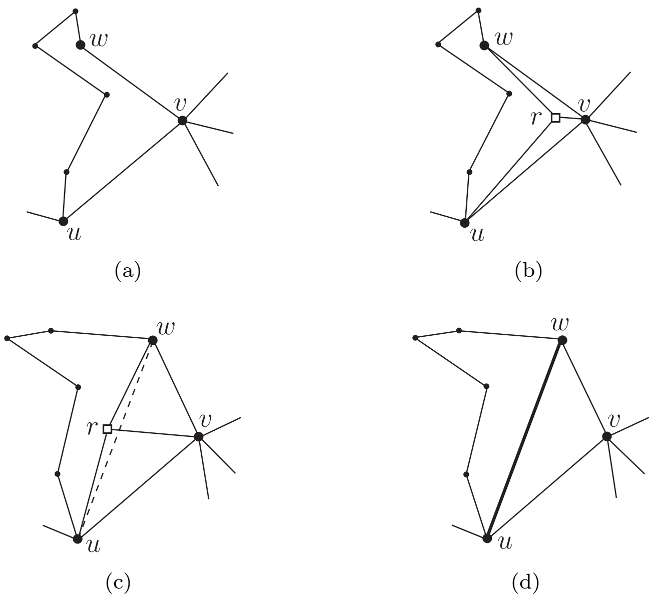

Fig. 3. *Morphing to add edge uw.* (a) *Vertices u, v, and w are consecutive around a facial walk, and the graph does not currently contain the edge uw.* (b) *A vertex r is added suitably close to v and connected to v, u, and w.* (c) *After convexifying the quadrilateral urwv.* (d) *Vertex r and its incident edges can be removed in order to insert edge uw.*

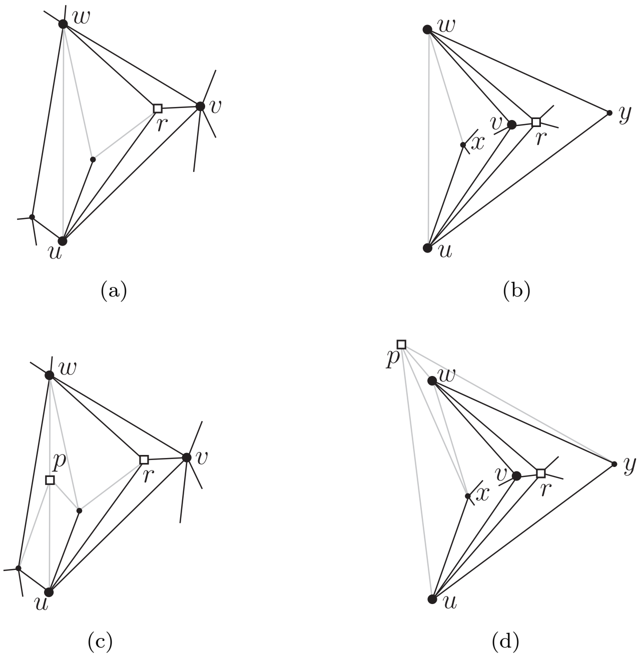

Fig. 4. *If edge uw has been added when triangulating the drawing, as in* (a) *and* (b)*, then we remove uw, add a new vertex p, and add straight-line edges from p to the vertices of the quadrilateral resulting from the removal of uw, as in* (c) *and* (d). *Drawings* (a) *and* (c) *illustrate the case in which uw is an internal edge of the triangulation, while drawings* (b) *and* (d) *illustrate the case in which uw is an external edge.*

After convexifying the quadrilateral $urwv$, we remove the temporary triangulation edges, remove r, and add the edge uw. See Figure 3(d).

Thus with two calls to Quadrilateral Convexification (one for each drawing), we have added one edge to both drawings. Moreover, the two drawings are still topologically equivalent drawings of the same planar graph. We can continue until every facial cycle of every connected component (except an isolated vertex or edge) is a triangle. Since planar graphs have $O(n)$ edges, in total we use $O(n)$ calls to Quadrilateral Convexification.

Part B. At this point every connected component of 3 or more vertices has facial cycles that are triangles. We will now morph and add edges to connect the components together, keeping the two drawings topologically equivalent. Once all of the components have been connected we will again appeal to Part A to triangulate the full connected graph.

We will handle the internal faces one by one. Consider one internal face F. The outer boundary of F is a triangle abc, and F has some number, c', of inner boundaries, each of which is a triangle, vertex, or edge. Each inner boundary that is a triangle has a subgraph (possibly disconnected) drawn inside it, which we call an "inner subgraph." Our goal is to add c' edges inside F to connect each inner boundary with the outer boundary. The high-level idea is to shrink the drawings of all of the inner subgraphs so that there is freedom to move them around within abc. We will use this freedom to line up the drawings of the inner subgraphs in the same order in both drawings near edge ac, starting from a position near a. It will then be straightforward to add an edge from each inner subgraph to vertex b. This will complete the processing of face F.

We remark that the reason for applying Part A before Part B is so that we can shrink and move the drawing of each inner subgraph using Lemma 2.1, which assumes a drawing inside a triangle.

For the remainder of this section, we will write "inner subgraph" rather than "drawing of inner subgraph."

In order to determine the size to which we must shrink the inner subgraphs, it will be helpful to have triangular boundaries for each inner subgraph (even for the ones that are single vertices or edges) that are disjoint, lie within abc, and have positive area.

To this end, consider each inner subgraph that is a vertex u. In each drawing we can insert a sufficiently small positive-area triangle formed by vertex u and by two new "dummy" vertices, so that it contains no part of the drawing other than vertex u itself. Similarly, consider each inner subgraph that is an edge uv. In each drawing we can insert a sufficiently thin positive-area triangle formed by edge uv and by a new "dummy" vertex, so that is contains no part of the drawing other than edge uv itself. In both cases the triangle becomes the triangular boundary for that subgraph.

Now, consider a uniform triangular grid such that a lies on a vertex of the grid and each grid cell is either a homothetic copy of abc (called an *upward* cell) or a homothetic copy of the triangle obtained by rotating abc by π radians (called a *downward* cell). Let every second row of cells in any of the three directions determined by the sides of abc be a *road*. Let any cell that is not in any road be a *home*. We fix the parities of the rows such that the upward cell adjacent to a in abc is a home. Note that all homes are thus upward cells. See Figure 5. We have yet to specify the size of a grid cell. We require a grid size such that the following conditions are satisfied: (1) there are at least c' homes along ac (recall that c' is the number of inner subgraphs of face F) and (2) each bounding triangle of an inner subgraph contains a home.

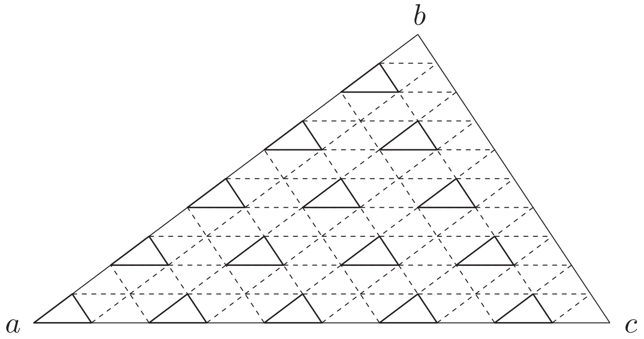

FIG. 5. *Roads (bounded by dashed lines) and homes (small solid triangles) within triangular face abc.*

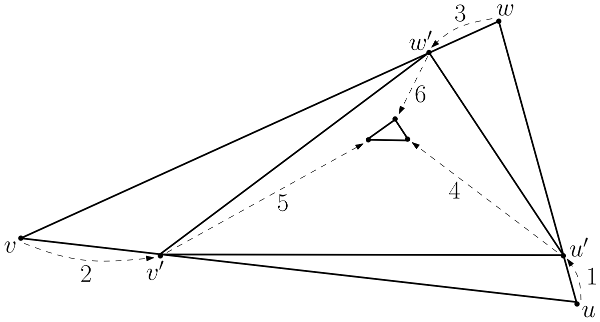

FIG. 6. *Movements of the vertices u, v, and w used to shrink uvw into h; the movements are labeled by the order in which they are performed. The triangles are, from outermost to innermost, uvw, u′v′w′, and the final position of uvw within h.*

For each inner subgraph, let its *home region* be the largest homothetic copy of an upward cell that can be inscribed in the subgraphs's bounding triangle. Let d be the diameter of abc. We can, for example, set the grid size such that the diameter of a grid cell is the minimum of (1) $d/(2c')$ and (2) a fifth of the diameter of the smallest home region. If the diameter of a grid cell is at most the first value, then there must be at least $2c'$ upward cells along ac, half of which are homes. If the diameter of a grid cell is at most the second value, then every home region is a (translated) copy of an upward cell, uniformly scaled by a factor of at least five. Every home region of this size must contain a home. In fact, every home region of this size contains a homothetic copy of abc whose side lengths are three times the side lengths of an upward cell; further, at least one of the six upward cells contained in this homothetic copy of abc is a home.

Consider an inner subgraph bounded by triangle uvw with a home region that contains a home h; our goal is to morph the inner subgraph into h. We will do so with $O(1)$ linear morphs; in each of them one vertex among u, v, and z moves linearly, while the other two stay put, and the vertices inside triangle uvw move along linearly. By Lemma 2.1, these are each unidirectional morphs. Refer to Figure 6.

Let $u'v'w'$ be the home region of uvw. We first morph uvw to coincide with $u'v'w'$ with a linear movement of each of u, v, and w. Then, with a linear movement of each vertex, we shrink uvw into h such that uvw becomes a slightly smaller homothetic copy of h and does not intersect the boundary of h. There is no interference between

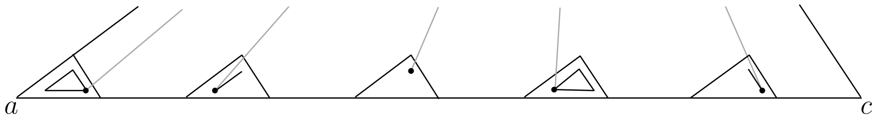

FIG. 7. *Adding edges (represented only partially) between the connectors and b. Connectors are represented by black disks.*

inner subgraphs since uvw and its contents remain in its original bounding triangle for the duration of the morph. Observe that we performed $O(1)$ unidirectional morphs for each of the c' inner subgraphs.

We can now arrange the c' inner subgraphs in any arbitrary order in the homes along edge ac using c' home swaps (each home swap moves one inner subgraph into the home previously occupied by another inner subgraph, and vice versa). Since all of the inner subgraphs are in homes, the road network is unimpeded and the roads are wide enough to allow passage of any inner subgraph. By navigating along the road network, it is straightforward to swap the homes of two inner subgraphs using $O(1)$ unidirectional morphs of the inner subgraphs. Thus, in $O(c')$ unidirectional morphs, we can arrange the inner subgraphs into the homes along ac in the same order in both drawings. Finally, in order to connect all the inner subgraphs to the outer boundary triangle abc, we choose a nondummy boundary vertex for each of them and call it the subgraph's *connector*; this choice has to be done in the same way in both drawings. We add edges between all the connectors and b, as in Figure 7. Note that these new edges do not cross each other or the inner subgraphs. This completes the processing of face F.

Once all inner faces F have been handled as above, the graph is connected. We now remove all dummy vertices and edges and appeal to Part A to triangulate the full connected graph. This completes the description of Part B.

Altogether, we use $O(c)$ unidirectional morphs to connect the c connected components of the graph. We also add $O(n)$ edges in Part A, and each addition takes two calls to quadrilateral convexification. Furthermore, each call to Quadrilateral Convexification uses $O(1)$ unidirectional morphing steps by Theorem 3.2. Therefore, in total we use $O(n)$ unidirectional morphing steps.

We now analyze the running time of the algorithm. We call Quadrilateral Convexification $O(n)$ times, and each call takes $O(n^2)$ time. This dominates the other work done by the algorithm (e.g., we find $O(n)$ temporary triangulations, each of which takes $O(n \log n)$ time). Thus the total running time is $O(n^3)$. □

5. Morphing between two triangulations.

In this section we prove our main result, Theorem 1.1, for the case of triangulations. Since the previous section reduced the general case to the case of triangulations, this will complete the proof of Theorem 1.1.

THEOREM 5.1. *Let Γ_1 and Γ_2 be two triangulations that are topologically equivalent drawings of an n-vertex maximal planar graph G. There is a morph from Γ_1 to Γ_2 that is composed of $O(n)$ unidirectional morphs. More precisely, the morph uses $O(n)$ unidirectional morphs plus $O(n)$ calls to Quadrilateral Convexification. Furthermore, the morph can be constructed in $O(n^3)$ time.*

As described in subsection 1.1, our approach is to find an internal vertex v of degree at most five and then morph the drawings using $O(1)$ unidirectional morphs so that v can be contracted to the same neighbor u in both drawings. We will prove a time bound of $O(n^2)$ for this step. More details are given below.

After this step, we contract v to u in both drawings and apply recursion to find a morph $\mathcal{M}$ between the two smaller triangulations. The result is a pseudomorph: we contract v to u in Γ_1, apply $\mathcal{M}$, and then reverse the contraction of v to u in Γ_2. The number of steps satisfies the recurrence relation $S(n) = S(n-1) + O(1)$, which solves to $S(n) \in O(n)$. Finally, we apply Theorem 3.3 to convert the pseudomorph to a morph with the same number of unidirectional morphs. Each appeal to Theorem 3.3 takes time $O(n)$ since that is the bound on the number of unidirectional morphs. Thus the total time bound is given by the recurrence relation $T(n) = T(n-1) + O(n^2)$, which solves to $T(n) \in O(n^3)$.

We now fill in the details about finding vertex v and contracting it to a common neighbor in both drawings. If there are no internal vertices, then we only have the outer triangle and a morph with $O(1)$ unidirectional morphing steps is easily computed in time $O(1)$. Otherwise, we claim that there is an internal vertex v of degree at most 5. To see this, note that the triangulation has $3n-6$ edges, and thus the sum of the degrees is $6n-12$. Every vertex has degree at least 3, and if all the internal vertices had degree at least 6, the sum of the degrees would be at least $3(3)+6(n-3) = 6n-9$.

If v has degree 3 then we can contract v to the same neighbor u in both drawings. If v has degree 4 or 5 then let u be a neighbor of v to which v can be contracted in Γ_2. (The existence of u is guaranteed by applying Lemma 2.2 to the polygon $\Delta(v)$ in Γ_2.) If v can be contracted to u in Γ_1 we are done. Otherwise, we will morph Γ_1 to make this possible. Note that there are no external chords from u to any vertex of $\Delta(v)$, because the two drawings are topologically equivalent, and Γ_2 has no such chords.

If v has degree 4 then the neighbors of v must form a nonconvex quadrilateral Q with vertices $abcd$, where d, say, is the reflex vertex. We contract v to d and apply Quadrilateral Convexification to Q, noting that the quadrilateral has no external chords. Specifically, there is no chord ac because v cannot be contracted to u, so u must be one of a, c, and, as noted above, there are no external chords incident to u. After convexification, we move v slightly into the interior of $\Delta(v)$ to obtain a drawing in which vertex v can be contracted to u. The result is a pseudomorph, which we convert to a morph using Theorem 3.3. The number of unidirectional steps is $O(1)$ and the time bound is $O(n^2)$ by Theorem 3.2.

If v has degree 5 then the neighbors of v form a pentagon $abcde$ where we want to contract v to $u = a$, say. We use a similar method, morphing the pentagon until it is "almost" convex by making a few calls to Quadrilateral Convexification. More formally, we have the following.

Lemma 5.2. *Let Γ be an n-vertex triangulation. Suppose v is an internal vertex of degree* 5*, with neighbors forming a pentagon $abcde$, and suppose that a is not incident to any external chords of the pentagon. Then we can morph Γ, keeping the outer boundary fixed, so that v can be contracted to a. The morph consists of at most two calls to Quadrilateral Convexification plus a constant number of unidirectional morphing steps. Furthermore, the morph can be found in $O(n^2)$ time.*

Proof. Because v's neighbors form a pentagon, v can be contracted to some neighbor x while preserving planarity (Lemma 2.2). If $x = a$ we are done, so suppose they are different. Contract v to x. We will divide into two cases depending on whether x is adjacent to a on the pentagon or not.

First consider the case where x is not adjacent to a. Without loss of generality, suppose that $x = c$. See Figure 8(a). We use Quadrilateral Convexification to convexify $acde$. This is possible since ce is an inner chord and ad is not an external chord.

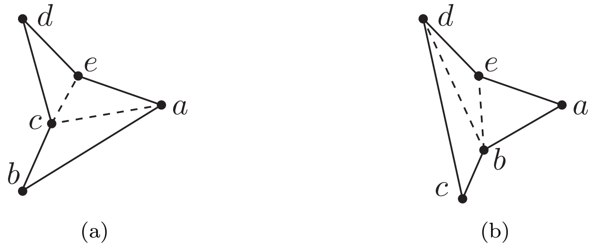

FIG. 8. (a) *Vertex v has been contracted to $x = c$ and we wish to contract v to a.* (b) *Vertex v has been contracted to $x = b$ and we wish to contract v to a.*

Now a sees all the quadrilateral *acde* as well as the triangle *abc*. Thus v can now be contracted to a.

Next consider the case where x is adjaceqt to a. Without loss of generality, suppose that $x = b$. See Figure 8(b). Use Quadrilateral Convexification to convexify *abde*. This is possible since *be* is an inner chord and *ad* is not an external chord. Now d sees all the quadrilateral *abde* as well as the triangle *bcd*. Move vertex v in a straight line from $x = b$ to d. This puts us in the first case. We use Theorem 3.3 to convert the resulting pseudomorph to a morph.

The morph consists of at most two calls to Quadrilateral Convexification plus a constant number of unidirectional morphing steps. The morph can be found in $O(n^2)$ time. □

6. Quadrilateral convexification.

In this section we give an algorithm for Problem 3.1, to morph a triangulation in order to convexify a quadrilateral. The main result of the section is as follows.

Reminder of Theorem 3.2. Quadrilateral Convexification can be solved via a single unidirectional morph. Furthermore, such a morph can be found in $O(n^2)$ time.

In the rest of this section we will prove Theorem 3.2. Recall that we have an n-vertex triangulation Γ and a quadrilateral *abcd* in Γ with no vertex inside it and such that neither *ac* nor *bd* is an edge outside of *abcd*. Our goal is to morph Γ so that *abcd* becomes convex using a single unidirectional morph.

We first describe the main idea. If *abcd* is convex, then we are done. Assume without loss of generality that d is the reflex vertex of the nonconvex quadrilateral, i.e., the angle at d internal to quadrilateral *abcd* is larger than π radians. This implies that b is the tip of the arrowhead shape and that the triangulation contains the edge *bd* (see Figure 9). Change the frame of reference so that *bd* is almost horizontal. We will use one unidirectional morph that moves vertices along horizontal lines, i.e., preserving their y-coordinates. Our main tool will be a result about redrawing a plane graph to have convex faces while keeping all vertices at the same y-coordinate—this is a result by Hong and Nagamochi [27] expressed in terms of level planar drawings of hierarchical graphs. To complete the proof we will show that the linear motion from the original drawing to the new drawing in fact preserves planarity, and therefore is a unidirectional morph.

We introduce some definitions and terminology that are needed for Hong and Nagamochi's result. A hierarchical graph is a graph with vertices assigned to layers. More formally, a *hierarchical graph* is a triple (G, L, γ) such that G is a graph, L is a set of horizontal lines (sometimes called *layers*), and γ is a function mapping each

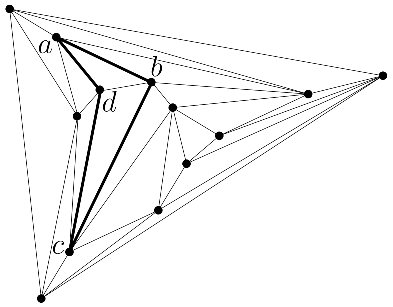

FIG. 9. *Triangulation* Γ*, containing a nonconvex quadrilateral abcd with a reflex vertex d, with no vertex inside it, with edge bd inside it, and with no chord outside it.*

vertex of G to a line in L in such a way that, if an edge (u, v) belongs to G, then $\gamma(u) \neq \gamma(v)$. (It is conventional to assume the layers are $1, 2, \dots k$, i.e., that successive horizontal lines are distance 1 apart, but that turns out to be unnecessary for Hong and Nagamochi's result, as the fact that the horizontal lines are equally spaced is nowhere used in their proof.) With the lines in L ordered from bottom to top, γ represents a partial order on the vertices of G, and we write $u \prec_\gamma v$ if the line $\gamma(u)$ is below the line $\gamma(v)$.

A *level drawing* of a hierarchical graph (G, L, γ) maps each vertex v of G to a point on the line $\gamma(v)$ and each edge to a y-monotone curve. A *level planar drawing* is a level drawing in which no two curves representing edges cross; such a drawing is *straight-line* if edges are drawn as straight-line segments, and *convex* if faces are delimited by convex polygons.

In our situation, we have a straight-line level planar drawing of a hierarchical graph and we want a straight-line convex level planar drawing that "respects" the embedding. More abstractly and more generally, Hong and Nagamochi define a *hierarchical plane graph* to be a hierarchical graph together with a combinatorial embedding (a rotation system and outer face) corresponding to some level planar drawing. In other words, a hierarchical plane graph is an equivalence class of drawings, and a drawing of a hierarchical plane graph must be a member of the equivalence class.

In order to guarantee a convex level planar drawing, Hong and Nagamochi require some conditions. Given a hierarchical plane graph (G, L, γ), an *st-face* of G is a face delimited by two paths $(s = u_1, u_2, \dots, u_k = t)$ and $(s = v_1, v_2, \dots, v_l = t)$ such that $u_i \prec_\gamma u_{i+1}$ for every $1 \leq i \leq k-1$, and such that $v_i \prec_\gamma v_{i+1}$ for every $1 \leq i \leq l-1$. We say that (G, L, γ) is a *hierarchical plane st-graph* if every face of G is an st-face. Hong and Nagamochi give an algorithm that constructs a convex straight-line level planar drawing of any hierarchical plane st-graph [27]. Here we explicitly formulate a weaker version of their main theorem.[2]

THEOREM 6.1 (see Hong and Nagamochi [27]). *Every* 3*-connected hierarchical plane st-graph* (G, L, γ) *admits a convex straight-line level planar drawing.*

[2] We make some remarks. First, the result in [27] proves that a convex straight-line level planar drawing of (G, L, γ) exists even if a convex polygon representing the cycle delimiting the outer face of G is arbitrarily prescribed. More precisely, Hong and Nagamochi show a necessary and sufficient condition for a convex polygon (possibly with flat angles) to be the polygon delimiting the outer face of a convex straight-line level planar drawing of (G, L, γ). However, this condition is always satisfied if every internal angle of the polygon is convex. Second, the result holds for a superclass of the 3-connected planar graphs, namely, for all the graphs that admit a convex straight-line drawing [15, 43].

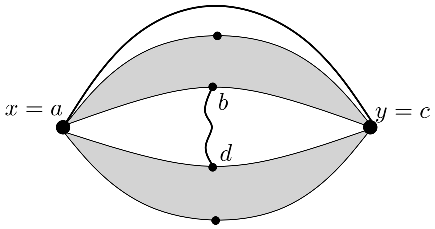

FIG. 10. *Illustration for the proof of Lemma* 6.2.

We now proceed to prove Theorem 3.2 and hence to solve the Quadrilateral Convexification problem with a single unidirectional morph. First, we rotate the frame of reference so that edge bd is horizontal and then we rotate it a bit more, so that no two vertices have the same y-coordinate and so that the strip delimited by the horizontal lines through b and d contains no vertex of Γ, except for the presence of b and d on its boundary. Remove edge bd from Γ obtaining a planar straight-line drawing Γ' of a planar graph G'. Draw a horizontal line through each vertex; let L be the set of these lines and let γ be the function that maps each vertex to the unique line in L through it; observe that, by the assumptions, γ represents a total order on the vertices of G'. We have the following.

LEMMA 6.2. *(G', L, γ) is a 3-connected hierarchical plane st-graph.*

Proof. By construction, Γ' is a straight-line level planar drawing of (G', L, γ), hence (G', L, γ) is a hierarchical plane graph.

Further, every face of G' is an st-face. This is trivially true for all faces delimited by triangles in Γ', since γ represents a total order on the vertices of G. Moreover, this is true for the face delimited by $abcd$, since by the choice of the reference frame a is below both horizontal lines $\gamma(b)$ and $\gamma(d)$ and c is above both of them, or vice versa. It follows that (G', L, γ) is a hierarchical plane st-graph.

Finally, we prove that G' is 3-connected. Refer to Figure 10. Suppose, for a contradiction, that G' is not 3-connected. Then there exist two vertices x and y whose removal from G' results in a disconnected graph G''. Now Γ is 3-connected since it is a triangulation, so adding the edge bd to G'' reconnects this graph. Therefore, b and d must be in different components of G'' (in particular, this implies that $\{x, y\} \cap \{b, d\} = \emptyset$). Since G' contains paths bad and bcd, it follows that $\{x, y\} = \{a, c\}$. Since removing a, c disconnects G', there is another face of G' that contains a and c. This face is delimited by three edges (as the one delimited by quadrilateral $abcd$ is the only face of G' which is not triangular), therefore contains the edge ac. However, this implies that $xy = ac$ is a chord external to polygon $abcd$ in Γ, thus contradicting the assumptions. □

By Lemma 6.2, (G', L, γ) is a 3-connected hierarchical plane st-graph. By Theorem 6.1, a convex straight-line level planar drawing Λ' of (G', L, γ) exists. In particular, polygon $abcd$ is convex in Λ'. Construct a straight-line planar drawing Λ of G from Λ' by drawing edge bd as an open straight-line segment. Due to the initial choice of the reference frame, this introduces no crossing in the drawing. To solve the Quadrilateral Convexification we use a single linear morph, $\langle\Gamma, \Lambda\rangle$, from Γ into Λ. See Figure 11. We now prove the following lemma.

LEMMA 6.3. *The linear morph $\langle\Gamma, \Lambda\rangle$ is planar and unidirectional.*

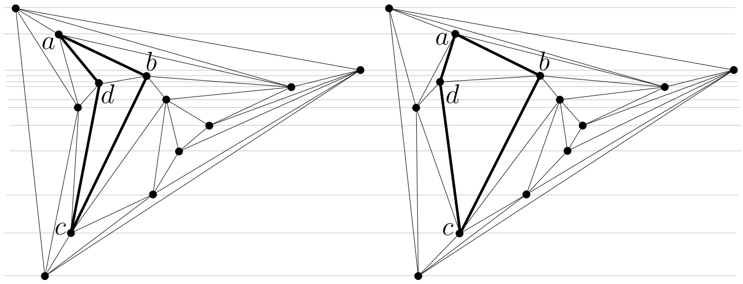

FIG. 11. *The linear morph* $\langle \Gamma, \Lambda \rangle$.

Proof. The morph is certainly unidirectional, since each vertex is on the same horizontal line in the initial drawing Γ and the final drawing Λ. To prove planarity we will argue that no vertex crosses an edge during the motion. The tool we need is Lemma 7.1, which proves that if two points p and q each move at uniform speed along a horizontal line and q is to the right of p in their initial and final positions, then q is to the right of p at every time instant. Lemma 7.1 will be stated and proved in section 7.2.1. Thus it suffices to show that for each horizontal line ℓ in L, the left-to-right ordering of vertices lying on ℓ and points where edges cross ℓ is the same in Γ as in Λ. This follows directly from the fact that both drawings have the same faces, and every face is an st-face. In particular, every horizontal line crosses an st-face at most twice, and the order of these crossings is the same because the cyclic order of edges around the face is the same. ◻

In order to complete the proof of Theorem 3.2, it remains to discuss the running time of the algorithm that solves Quadrilateral Convexification. Removing and reinserting edge bd takes $O(1)$ time. Other than that, we only need to apply Hong and Nagamochi's algorithm, which takes $O(n^2)$ time [27]. Thus the total running time is $O(n^2)$.

7. Converting a pseudomorph to a morph.

In this section, we show how to convert a pseudomorph consisting of unidirectional morphing steps into a true morph of unidirectional morphing steps, assuming that the graph is triangulated and the vertices we contract have degree at most 5. Specifically, we prove the following.

Reminder of Theorem 3.3. Let Γ_1 and Γ_2 be two triangulations that are topologically equivalent drawings of an n-vertex maximal planar graph G. Suppose that there is a pseudomorph from Γ_1 to Γ_2 in which we contract an internal vertex v of degree at most 5, perform k unidirectional morphs, and then uncontract v. Then there is a morph $\mathcal{M}$ from Γ_1 to Γ_2 that consists of $k+2$ unidirectional morphs. Furthermore, given the sequence of $k+1$ drawings that define the pseudomorph, we can modify them to obtain the sequence of drawings that define $\mathcal{M}$ in $O(k+n)$ time.

Suppose that the given pseudomorph consists of the contraction of an internal vertex v with $\deg(v) \leq 5$ to a vertex a, followed by a morph $\mathcal{M}' = \langle \Gamma_1, \ldots, \Gamma_{k+1} \rangle$ of the reduced graph, and then the uncontraction of v from a. Suppose that $\langle \Gamma_i, \Gamma_{i+1} \rangle$ is an L_i-directional morph for $1 \leq i \leq k$. We assume that the drawings $\Gamma_1, \ldots, \Gamma_{k+1}$ are given to us, and show how to update the sequence of drawings to those of $\mathcal{M}$ in time $O(n+k)$.

Specifically, we will show how to add v and its incident edges back into each drawing of the morph $\mathcal{M}'$ keeping each step unidirectional. We will preserve planarity by placing v at an interior point of the kernel of $\Delta(v)$. Call the resulting morph $\mathcal{M}''$. We will perform the modifications from $\mathcal{M}'$ to $\mathcal{M}''$ in time $O(k)$. To obtain the final morph $\mathcal{M}$, we replace the original contraction of v to a by a unidirectional morph that moves v from its initial position to its position at the start of $\mathcal{M}''$, then follow the steps of $\mathcal{M}''$, and then replace the uncontraction of v by a unidirectional morph that moves v from its position at the end of $\mathcal{M}''$ to its final position. The result is a true morph that consists of $k+2$ unidirectional morphing steps. It takes $O(n)$ time to add the two extra unidirectional morphs to the sequence, since we must add two drawings of an n-vertex graph.

Thus our main task is to modify the morph $\mathcal{M}'$ by adding vertex v and its incident edges back into each drawing of the morph sequence in constant time per drawing, preserving planarity and maintaining the property that each step of the morph sequence is unidirectional. We can ignore everything outside the polygon $P = \Delta(v)$. Note that, as vertex a is adjacent to all the vertices of P, it remains in the kernel of P throughout the morph $\mathcal{M}'$. We distinguish cases depending on the degree of v. Section 7.1 proves Theorem 3.3 for the easy case where the degree of v is 3 or 4. Section 7.2 proves Theorem 3.3 for the more complicated case where v has degree 5.

7.1. Vertex *v* of degree 3 or 4. In this section we prove Theorem 3.3 for the case where the contracted vertex v has degree 3 or 4, i.e., P is a triangle or quadrilateral. If P is a triangle then by Lemma 2.1 we can place v at a fixed convex combination of the triangle vertices in all the drawings Γ_i.

If P is a quadrilateral $abcd$ then the line segment ac stays in the kernel of P because vertex a stays in the kernel of P. Thus, we can place v at a fixed convex combination of a and c in all the drawings Γ_i (using the degenerate version of Lemma 2.1 where the triangle collapses to a line segment).

The coordinates of v in each Γ_i can be computed in constant time, so the total time bound is $O(k)$.

7.2. Vertex *v* of degree 5. In this section we prove Theorem 3.3 for the case where the contracted vertex v has degree 5. Our goal will be to place vertex v very close to the vertex a to which it was contracted in the pseudomorph. Let $P = \Delta(v)$ be the pentagon $abcde$ labeled clockwise. We use the notation that b is at point b_i in drawing Γ_i, etc.

We may assume that vertex a is fixed throughout the entire morph. This is not a loss of generality because if vertex a moves, we can translate the whole drawing to move it back: the morph in which every vertex v moves k_v units along direction $\bar{\ell}$ is planar, if and only if the morph in which every vertex moves $k_v - k_a$ units along direction $\bar{\ell}$ is planar, and note that vertex a stays fixed in the latter morph.

We want to place v within distance ε of a, with ε small enough so that at any time instant t during morph $\langle \Gamma_1, \ldots, \Gamma_{k+1} \rangle$ the intersection between the disk D centered at a with radius ε and the kernel of polygon P consists of a positive-area sector S of D. Since the morph consists of k linear morphs, we can compute a value for ε as follows. For $1 \leq i \leq k$ let ε_i be the minimum distance from a to any of the edges bc, cd, de during the unidirectional morph $\langle \Gamma_i, \Gamma_{i+1} \rangle$. We claim that ε_i can be computed in constant time on our real RAM model of computation. It suffices to compute the minimum distance between a and each of the three infinite lines through bc, cd, and de. This is because we can test if the minimum occurs inside the relevant line segment, and we can compute the minimum distance from a to each of the moving endpoints b, c, d, e. Consider the line through points p, q, where $pq = bc, cd$, or de, and points p

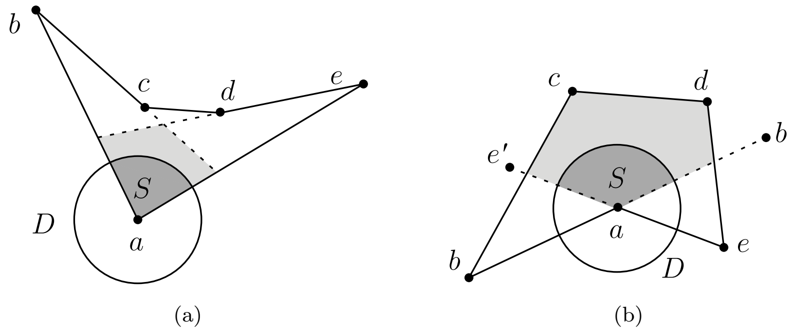

FIG. 12. *A disk D centered at a whose intersection with the kernel of P (the lightly shaded polygonal region) is a non-zero-area sector S (darkly shaded).* (a) *Vertex a is convex and S is a positive sector.* (b) *Vertex a is reflex and S is a negative sector.*

and q each move on parallel lines at uniform speed during the morph $\langle \Gamma_i, \Gamma_{i+1} \rangle$. As a function of time, t, the square of the distance from a to the line through pq is of the form $f(t)/g(t)$, where f and g are quadratic polynomials in t. We want to solve for the derivative being zero. The derivative has a cubic polynomial in the numerator, and a cubic equation can be solved in constant time on a real RAM model.

After computing each ε_i, let $\varepsilon = \min_{1 \le i \le k}\{\varepsilon_i\}$. Then ε can be computed in time $O(k)$.

Fix D to be the disk of radius ε centered at a. In case a is a convex vertex of P, the sector S is bounded by the line segments ab and ae and we call it a *positive* sector. See Figure 12(a). In case a is a reflex vertex of P, the sector S is bounded by the extensions of line segments ab and ae and we call it a *negative* sector. See Figure 12(b). More precisely, in case a is a reflex vertex of P, let b' and e' be points so that a is the midpoint of the segments bb' and ee', respectively; then the negative sector is bounded by the segments ae' and ab'. Note that when an L-directional morph is applied to P, the points b' and e' also move at uniform speed in direction L.

The important property we use from now on is that any point in the sector S lies in the kernel of polygon P. This property immediately follows from the choice of ε. Let the sector in drawing Γ_i be S_i, for $i = 1, \ldots, k+1$. Recall our convention that $\langle \Gamma_i, \Gamma_{i+1} \rangle$ is an L_i-directional morph.

Our task is to choose, for each $i = 1, \ldots, k+1$, a position v_i for vertex v inside sector S_i; this has to be done so that the L_j-directional morph $\langle \Gamma_j, \Gamma_{j+1} \rangle$ keeps v inside the sector at all times, for every $j = 1, \ldots, k$. We will separate the proof into two parts. One part is to show that there exist points v_i in S_i so that the line segment $v_i v_{i+1}$ is parallel to L_i. This is in section 7.2.2. The other part of the proof is to show that such points v_i ensure that v is inside sector S throughout the morph. This is in section 7.2.1.

For both proofs, we will distinguish the following two possibilities for the relationship between S_i and the line L_i translated to go through point a.

One-sided case. Points b_i and e_i lie in the same closed half-plane determined by L_i. In this case, whether the sector S_i is positive or negative, L_i does not intersect the interior of S_i. See Figure 13. An L_i-directional morph keeps b_i and e_i on the same side of L_i so if S_i is positive it remains positive and if S_i is negative it remains negative. Observe that v remains inside the sector if and only if it remains inside D and between the two half-lines ab and ae.

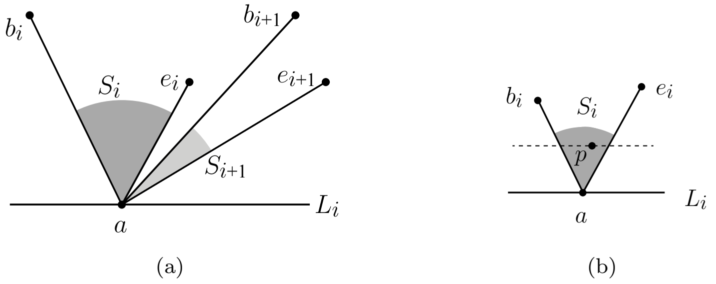

FIG. 13. *The one-sided case where S_i lies to one side of L_i, illustrated for a positive sector S_i.* (a) *An L_i-directional morph to S_{i+1}.* (b) *v remains inside the sector if and only if it remains inside D and between the two half-lines ab and ae.*

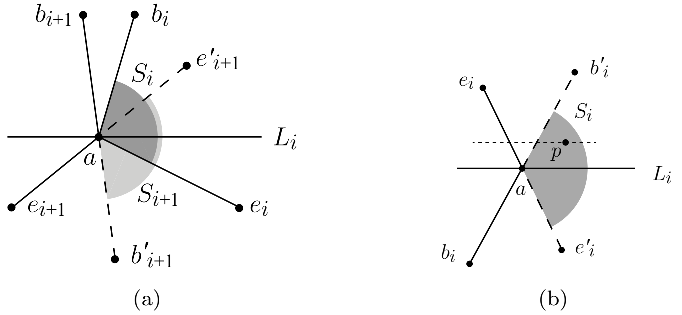

FIG. 14. *The two-sided case where S_i contains points on both sides of L_i.* (a) *An L_i-directional morph from the positive sector S_i bounded by b_iae_i to the negative sector S_{i+1} bounded by $e'_{i+1}ab'_{i+1}$.* (b) *v remains inside the sector if and only if it remains inside D and on the same side of the lines bb' and ee'.*

Two-sided case. Points b_i and e_i lie on opposite sides of L_i. In this case L_i intersects the interior of the sector S_i. See Figure 14. During an L_i-directional morph the sector S_i may remain positive, or it may remain negative, or it may switch between the two, although it can only switch once, due to Corollary 7.2 below. Observe that v remains inside the sector if and only if it remains inside D and on the same side of the lines bb' and ee'.

7.2.1. Sectors and the betweenness property. In this subsection we show that if we can choose points v_i in sector S_i such that the line v_iv_{i+1} is parallel to L_i for all $i \in \{1, \ldots, k\}$, then v remains inside the sector throughout the morph.

Our main tool is the following lemma proving that "sidedness" on line L is preserved in an L-directional morph.

LEMMA 7.1. *Let L be a horizontal line and p_0, p_1, q_0, q_1 be points in L. Consider a point p that moves at constant speed from p_0 to p_1 in one unit of time. If q_i is to the right of p_i, $i = 0, 1$, and q is a point that moves at constant speed from q_0 to q_1 in one unit of time, then q is to the right of p during their entire movement. Note that p_0 may lie to the right or left of p_1 and likewise for q_0 and q_1.*

Proof. Let p_i and q_i, $i = 0, 1$, be points as described above; see Figure 15. Denote by p_t and q_t the positions of p and q for $0 \leq t \leq 1$. First note that

$$q_i = p_i + \delta_i \tag{1}$$

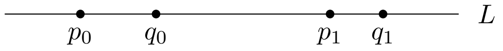

FIG. 15. *Points p and q move from p_0 to p_1 and from q_0 to q_1, respectively.*

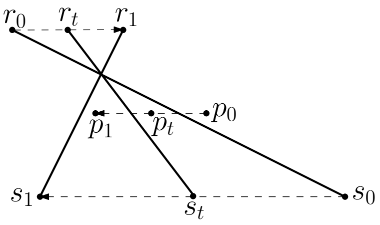

FIG. 16. *Illustration for the statement of Corollary 7.2. Points r_0 and r_1 denote the positions of r at times 0 and 1, respectively. Point r_t denotes the position of r at a generic time instant t such that $0 < t < 1$. Points s_0, s_1, s_t, p_0, p_1, and p_t are defined analogously.*

for $i = 0, 1$, with $\delta_i > 0$. Since p and q are moving at constant speed, we have $p_t = (1 - t)p_0 + tp_1$ and $q_t = (1 - t)q_0 + tq_1$. Now, using (1) in the expression for q_t we have

$$\begin{aligned} q_t &= (1 - t)(p_0 + \delta_0) + t(p_1 + \delta_1) \\ &= p_t + (1 - t)\delta_0 + t\delta_1, \end{aligned}$$

where $(1 - t)\delta_0 + t\delta_1 > 0$. □

Clearly, Lemma 7.1 generalizes to a directed line L that is not necessarily horizontal, if we interpret "to the right of" as "further in the direction of" L. We also have the following (see Figure 16).

COROLLARY 7.2. *Consider an L-directional morph acting on points p, r, and s. If p is to the right of the line through rs at the beginning and the end of the L-directional morph, then p is to the right of the line through rs throughout the L-directional morph.*

To prove Corollary 7.2, consider a line L' through p that is parallel to L and apply Lemma 7.1 to points p and q, where q is the intersection of L' and the line through r and s.

We now present our main result about the relative positions of points v_i and v_{i+1}.

LEMMA 7.3. *If point v_i lies in sector S_i, point v_{i+1} lies in sector S_{i+1}, and the line segment v_iv_{i+1} is parallel to L_i, then v stays inside the sector during the entire morph $\langle\Gamma_i, \Gamma_{i+1}\rangle$.*

Proof. First consider the one-sided case. Suppose S_i is a positive sector (the case of a negative sector is similar). Observe that v remains in the sector during an L_i-directional morph if and only if it remains in the disk D and remains between the half-lines ab and ae. See Figure 13(b). Because v_i and v_{i+1} both lie in D, the line segment between them also lies in the disk, and v remains in D throughout the morph. In the initial configuration, v_i lies between the half-lines ab_i and ae_i, and in the final configuration v_{i+1} lies between the half-lines ab_{i+1} and ae_{i+1}. Therefore, by Corollary 7.2, v remains between the half-lines throughout the L_i-directional morph. Thus v remains inside the sector throughout the morph.

Now consider the two-sided case. Observe that a point v remains in the sector during an L_i-directional morph if and only if it remains on the same side of the lines

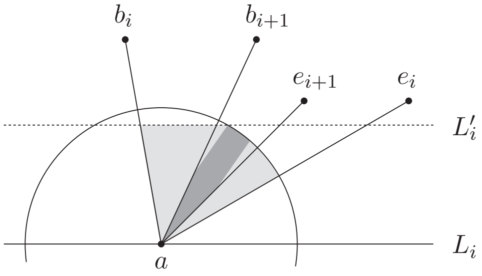

FIG. 17. *N_i (lightly shaded) is an L_i-truncation of S_i in the one-sided case. N_{i+1} is darkly shaded. The slab boundaries for N_i consist of L_i' and a parallel line just below L_i.*

bb' and ee'. Note that this is true even when the sector changes between positive and negative, since the points in S are those in D that are between the half-lines bb' and ee'. See Figure 14(b). As in the one-sided case, v remains in the disk throughout the morph. Also, v is on the same side of the lines bb' and ee' in the initial and final configurations and, therefore, by Corollary 7.2, v remains on the same side of the lines throughout the morph. Thus v remains inside the sector throughout the morph. □

7.2.2. Nice points. In this subsection we prove that there exist points v_i in S_i such that the line segment v_iv_{i+1} is parallel to L_i. We call the possible positions for v_i inside sector S_i the *nice* points, defined formally as follows:

- All points in the interior of S_{k+1} are nice.
- For $1 \leq i \leq k$, a point v_i in the interior of S_i is nice if there is a nice point v_{i+1} in S_{i+1} such that v_iv_{i+1} is parallel to L_i.

It suffices to show that all the sets of nice points are nonempty. We will in fact characterize the sets, and show how to efficiently compute nice points. Given a line L, an *L-truncation* of a sector S is the intersection of S with an open slab that is bounded by two lines parallel to L and contains a. Observe that this implies that the open slab will contain all points of S in a small neighborhood of a. In particular, an L-truncation of a sector is nonempty.

LEMMA 7.4. *The set of nice points in S_i is an L_i-truncation of S_i for $i = 1, \ldots, k$.*

Proof. Let N_i denote the nice points in S_i. The proof is by induction as i goes from $k+1$ to 1. All the points in the interior of S_{k+1} are nice. Suppose by induction that N_{i+1} is an L_{i+1}-truncation of S_{i+1}.

See Figure 17 for the one-sided case and Figure 18 for the two-sided case. The slab determining N_i consists of all lines parallel to L_i that go through a point of N_{i+1}. The slab is nonempty since N_{i+1} contains all of S_{i+1} in a small neighborhood of a. Thus the slab contains all points of S_i in a neighborhood of a, and thus N_i is an L_i-truncation of S_i. □

Lemma 7.4 implies, in particular, that the set of nice points is nonempty.

The last thing we need to do to complete the proof of Theorem 3.3 for the case where v has degree 5 is to show that the above procedure to compute the v_i's has a running time of $O(k)$. It suffices to show how to compute N_i from N_{i+1} in constant time. To do this, we just compute the maximum and minimum points of N_{i+1} in the direction perpendicular to L_i. The slab boundaries for N_i go through these points. Then N_i is the intersection of the slab with sector S_i.

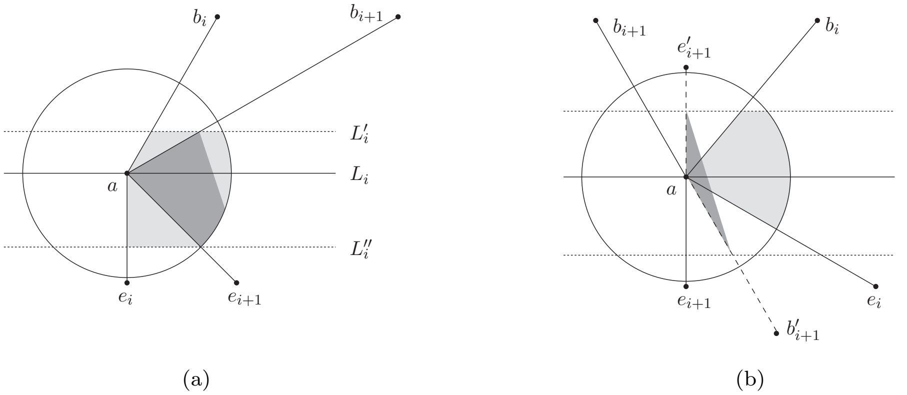

FIG. 18. *N_i (lightly shaded) is an L_i-truncation of S_i in the two-sided case. N_{i+1} is darkly shaded. L'_i and L''_i are the slab boundaries for N_i.*

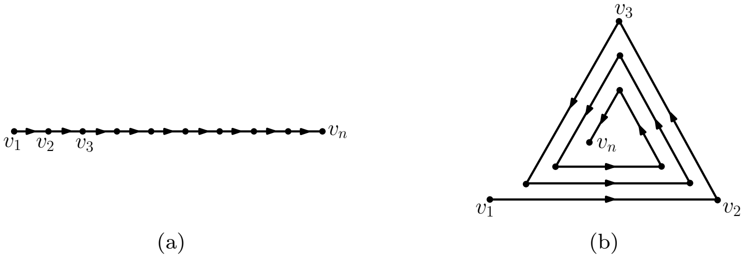

FIG. 19. *Drawings* Γ (a) *and* Λ (b).

8. A linear lower bound. In this section we prove Theorem 1.2, which states that there exist two straight-line planar drawings of an n-vertex path such that any straight-line planarity preserving morph between them that consists of k linear morphs is such that $k \in \Omega(n)$.

Specifically we will describe two straight-line planar drawings Γ and Λ of an n-vertex path $P = (v_1, \ldots, v_n)$, and prove that any straight-line planarity preserving morph $\mathcal{M}$ between Γ and Λ that consists of k linear morphs is such that $k \in \Omega(n)$. In order to simplify the description, we consider each edge $e_i = (v_i, v_{i+1})$ as oriented from v_i to v_{i+1} for $i = 1, \ldots, n-1$.

Drawing Γ (see Figure 19(a)) is such that all the vertices of P lie on a horizontal straight line with v_i to the left of v_{i+1} for each $i = 1, \ldots, n-1$.

Drawing Λ (see Figure 19(b)) is such that

- for each $i = 1, \ldots, n-1$ with $i \bmod 3 \equiv 1$, e_i is horizontal with v_i to the left of v_{i+1};
- for each $i = 1, \ldots, n-1$ with $i \bmod 3 \equiv 2$, e_i is parallel to line $y = \tan(\frac{2\pi}{3})x$ with v_i to the right of v_{i+1}; and
- for each $i = 1, \ldots, n-1$ with $i \bmod 3 \equiv 0$, e_i is parallel to line $y = \tan(-\frac{2\pi}{3})x$ with v_i to the right of v_{i+1}.

Let $\mathcal{M} = \langle \Gamma = \Gamma_1, \ldots, \Gamma_{k+1} = \Lambda \rangle$ be any straight-line planarity-preserving morph transforming Γ into Λ and consisting of k linear morphs.

For $i = 1, \ldots, n$ and $j = 1, \ldots, k+1$, we denote by v_i^j the point where vertex v_i is placed in Γ_j; further, $i = 1, \ldots, n-1$ and $j = 1, \ldots, k+1$, we denote by e_i^j the directed straight-line segment representing edge e_i in Γ_j.

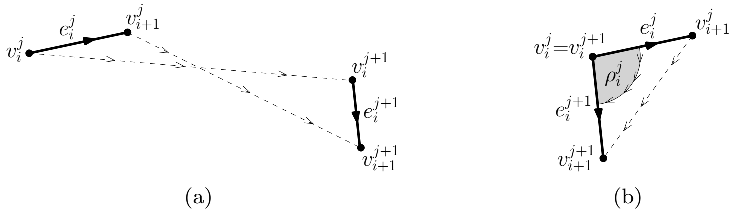

FIG. 20. (a) *Morph between* e_i^j *and* e_i^{j+1}. (b) *Translation of the positions of* e_i *during* $\langle \Gamma_j, \Gamma_{j+1} \rangle$, *resulting in* e_i *spanning an angle* ρ_i^j *around* v_i.

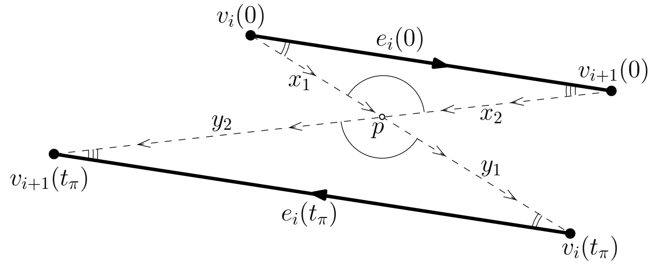

FIG. 21. *Illustration for the proof of Lemma* 8.1.

For $i = 1, \ldots, n-1$ and $j = 1, \ldots, k$, we define the *rotation* ρ_i^j of e_i around v_i during the morphing step $\langle \Gamma_j, \Gamma_{j+1} \rangle$ as follows (see Figure 20). Assume that vertex v_i is fixed throughout the morphing step $\langle \Gamma_j, \Gamma_{j+1} \rangle$ (that is, assume that v_i^j and v_i^{j+1} coincide); as observed in section 7.2, this is not a loss of generality. Then the morph between e_i^j and e_i^{j+1} is a rotation of e_i around v_i (where e_i might vary its length during $\langle \Gamma_j, \Gamma_{j+1} \rangle$) spanning an angle ρ_i^j. We assume $\rho_i^j > 0$ if the rotation is counterclockwise and $\rho_i^j < 0$ otherwise. We have the following.

LEMMA 8.1. *For each* $i = 1, \ldots, n-1$ *and* $j = 1, \ldots, k$, *we have* $|\rho_i^j| < \pi$.

Proof. Assume, for a contradiction, that $|\rho_i^j| \geq \pi$ for some $1 \leq i \leq n-1$ and $1 \leq j \leq k$. Also assume, without loss of generality, that the morphing step $\langle \Gamma_j, \Gamma_{j+1} \rangle$ happens between time instants $t = 0$ and $t = 1$.

We introduce some notation. For any $0 \leq t \leq 1$, we denote

- by $v_i(t)$ the position of v_i at time instant t—note that $v_i(0) = v_i^j$ and $v_i(1) = v_i^{j+1}$;
- by $v_{i+1}(t)$ the position of v_{i+1} at time instant t—note that $v_{i+1}(0) = v_{i+1}^j$ and $v_{i+1}(1) = v_{i+1}^{j+1}$;
- by $e_i(t)$ the drawing of e_i at time instant t—note that $e_i(0) = e_i^j$ and $e_i(1) = e_i^{j+1}$; and
- by $\rho_i^j(t)$ the rotation of e_i around v_i during the linear morph transforming $e_i(0)$ into $e_i(t)$—note that $\rho_i^j(0) = 0$ and $\rho_i^j(1) = \rho_i^j$.

Since a morph is a continuous transformation and since $|\rho_i^j| \geq \pi$, it follows that there exists a time instant t_π with $0 < t_\pi \leq 1$ such that $|\rho_i^j(t_\pi)| = \pi$. We prove that there exists a time instant t_r with $0 < t_r \leq t_\pi$ in which $v_i(t_r)$ and $v_{i+1}(t_r)$ coincide, thus contradicting the assumption that morph $\langle \Gamma_j, \Gamma_{j+1} \rangle$ is planar. Refer to Figure 21.

Since $|\rho_i^j(t_\pi)| = \pi$, it follows that $e_i(t_\pi)$ is parallel to $e_i(0)$ and oriented in the opposite way. This easily implies that t_r exists if $e_i(t_\pi)$ and $e_i(0)$ lie on a common straight line. Otherwise, the straight-line segments $v_i(0)v_i(t_\pi)$ and $v_{i+1}(0)v_{i+1}(t_\pi)$ meet in a point p. Let x_1, x_2, y_1, and y_2 be the lengths of the straight-line segments $pv_i(0)$, $pv_{i+1}(0)$, $pv_i(t_\pi)$, and $pv_{i+1}(t_\pi)$, respectively. By the similarity of triangles $v_i(0)pv_{i+1}(0)$ and $v_i(t_\pi)pv_{i+1}(t_\pi)$, we have $\frac{x_1}{y_1} = \frac{x_2}{y_2}$ and hence $\frac{x_1}{x_1+y_1} = \frac{x_2}{x_2+y_2}$. Thus, at time instant $t_r = \frac{x_1}{x_1+y_1} t_\pi$, we have that $v_i(t_r)$ and $v_{i+1}(t_r)$ coincide (in fact they both lie at p). This contradiction proves the lemma. □

For $j = 1, \dots, k$, we denote by $\mathcal{M}_j$ the morph $\langle \Gamma_1, \dots, \Gamma_{j+1} \rangle$—note that $\mathcal{M}_k = \mathcal{M}$; also, for $i = 1, \dots, n-1$, we define the *total rotation* $\rho_i(\mathcal{M}_j)$ of edge e_i around v_i during morph $\mathcal{M}_j$ as $\rho_i(\mathcal{M}_j) = \sum_{m=1}^{j} \rho_i^m$. Observe that the total rotation might be a value larger than 2π radians or smaller than 0 radians; that is, the sum is not taken modulo 2π.

We will show in Lemma 8.3 that there exists an edge e_i, for some $1 \le i \le n-1$, whose total rotation during the entire morph is linear in the size of the path; that is, $\rho_i(\mathcal{M}_k) = \rho_i(\mathcal{M}) \in \Omega(n)$. In order to do that, we first analyze the relationship between the total rotation of two consecutive edges of P.

LEMMA 8.2. *For each $j = 1, \dots, k$ and for each $i = 1, \dots, n-2$, we have that $|\rho_{i+1}(\mathcal{M}_j) - \rho_i(\mathcal{M}_j)| < \pi$.*

Proof. Suppose, for a contradiction, that $|\rho_{i+1}(\mathcal{M}_j) - \rho_i(\mathcal{M}_j)| \ge \pi$ for some $1 \le j \le k$ and $1 \le i \le n-2$. Assume that j is minimal under this hypothesis.

Since each vertex moves continuously during $\mathcal{M}_j$, there exists an intermediate drawing Γ^* of P, occurring during morphing step $\langle \Gamma_j, \Gamma_{j+1} \rangle$ because of the minimality of j, such that $|\rho_{i+1}(\mathcal{M}^*) - \rho_i(\mathcal{M}^*)| = \pi$, where $\mathcal{M}^* = \langle \Gamma_1, \dots, \Gamma_j, \Gamma^* \rangle$ is the morph obtained by concatenating $\mathcal{M}_{j-1}$ with the morphing step transforming Γ_j into Γ^*.

Recall that in Γ_1 edges e_i and e_{i+1} lie on the same straight line and have the same orientation. Then, since $|\rho_{i+1}(\mathcal{M}^*) - \rho_i(\mathcal{M}^*)| = \pi$, in Γ^* edges e_i and e_{i+1} are parallel and have opposite orientations. Also, since edges e_i and e_{i+1} share vertex v_{i+1}, they lie on the same line. This implies that such edges overlap, contradicting the hypothesis that $\mathcal{M}^*$, $\mathcal{M}_j$, and $\mathcal{M}$ are planar. □

We now prove the key lemma for the lower bound.

LEMMA 8.3. *There exists an index i such that $|\rho_i(\mathcal{M})| \in \Omega(n)$.*

Proof. Refer again to Figure 19. For every $1 \le i \le n-2$, edges e_i and e_{i+1} form an angle of π radians in Γ, while they form an angle of $\frac{\pi}{3}$ radians in Λ. Hence, the total rotation of e_{i+1} during the entire morph $\mathcal{M}$ has to be larger than the one of e_i by $\frac{2\pi}{3}$ radians (plus any multiple of 2π); that is, $\rho_{i+1}(\mathcal{M}) = \rho_i(\mathcal{M}) + \frac{2\pi}{3} + 2z_i\pi$, for some $z_i \in \mathbb{Z}$.

In order to prove the lemma, it suffices to prove that $z_i = 0$ for every $i = 1, \dots, n-2$. Namely, in this case, $\rho_{i+1}(\mathcal{M}) = \rho_i(\mathcal{M}) + \frac{2\pi}{3}$ for every $i = 1, \dots, n-2$, and hence $\rho_{n-1}(\mathcal{M}) = \rho_1(\mathcal{M}) + \frac{2\pi}{3}(n-2)$. This implies $|\rho_{n-1}(\mathcal{M}) - \rho_1(\mathcal{M})| \in \Omega(n)$, and thus $|\rho_1(\mathcal{M})| \in \Omega(n)$ or $|\rho_{n-1}(\mathcal{M})| \in \Omega(n)$.

Assume, for a contradiction, that $z_i \ne 0$, for some $1 \le i \le n-2$. If $z_i > 0$, then $\rho_{i+1}(\mathcal{M}) \ge \rho_i(\mathcal{M}) + \frac{8\pi}{3}$; further, if $z_i < 0$, then $\rho_{i+1}(\mathcal{M}) \le \rho_i(\mathcal{M}) - \frac{4\pi}{3}$. Since each of these inequalities contradicts Lemma 8.2, the lemma follows. □

We are now ready to prove Theorem 1.2. Namely, consider the two drawings Γ and Λ of path $P = (v_1, \dots, v_n)$ illustrated in Figure 19. By Lemma 8.3, there exists an edge e_i of P, for some $1 \le i \le n-1$, such that $|\sum_{j=1}^{k} \rho_i^j| \in \Omega(n)$. Since, by

Lemma 8.1, we have that $|\rho_i^j| < \pi$ for each $j = 1, \ldots, k$, it follows that $k \in \Omega(n)$. This concludes the proof of the theorem.

9. Conclusion.

In this paper we have given an $O(n^3)$-time algorithm that takes as input two straight-line planar drawings Γ_1 and Γ_2 of the same n-vertex planar graph with the same embedding, and finds a morph consisting of $O(n)$ unidirectional morphing steps from Γ_1 to Γ_2 that preserves straight-line planarity. We have also proved that straight-line planarity preserving morphs consisting of linear morphs require a linear number of steps, in the worst case.

Our algorithm works under the real RAM model of computation. However, we have not bounded the coordinate values used in our morphs, and it seems that they may require a superlogarithmic number of bits (though they can be described using a polynomial number of arithmetic operations). Consequently, the intermediate drawings produced by our morph may have an exponential ratio of the distances between the closest and farthest pairs of vertices and are not likely to be visually appealing. We leave as an open problem to find a morph that uses a polynomial number of discrete morphing steps and uses only a logarithmic number of bits per coordinate. Barrera-Cruz, Haxell, and Lubiw [10] made a first step in this direction by solving the case where the two drawings are Schnyder drawings.

Acknowledgments. Some of the work was done as part of an Algorithms Problem Session at the University of Waterloo, and we thank the other participants for helpful discussions. We thank Therese Biedl for a good suggestion about 3-connected graphs.

REFERENCES

[1] O. Aichholzer, G. Aloupis, E. D. Demaine, M. L. Demaine, V. Dujmovic, F. Hurtado, A. Lubiw, G. Rote, A. Schulz, D. L. Souvaine, and A. Winslow, *Convexifying polygons without losing visibilities*, in Proceedings of the 23rd Canadian Conference on Computational Geometry (CCCG '11), Toronto, Ontario, 2011, pp. 229–234.

[2] S. Alamdari, P. Angelini, T. M. Chan, G. Di Battista, F. Frati, A. Lubiw, M. Patrignani, V. Roselli, S. Singla, and B. T. Wilkinson, *Morphing planar graph drawings with a polynomial number of steps*, in Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '13), S. Khanna, ed., SIAM, Philadelphia 2013, pp. 1656–1667.

[3] M. Alexa, D. Cohen-Or, and D. Levin, *As-rigid-as-possible shape interpolation*, in Proceedings of the 27th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00), ACM, New York, 2000, pp. 157–164.

[4] P. Angelini, G. Da Lozzo, G. Di Battista, F. Frati, M. Patrignani, and V. Roselli, *Morphing planar graph drawings optimally*, in Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP '14), J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias, eds., Lecture Notes in Comput. Sci. 8572, Springer, Heidelberg, 2014, pp. 126–137.

[5] P. Angelini, G. Da Lozzo, F. Frati, A. Lubiw, M. Patrignani, and V. Roselli, *Optimal morphs of convex drawings*, in Proceedings of the 31st International Symposium on Computational Geometry (SoCG 2015), L. Arge and J. Pach, eds., LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015, pp. 126–140.

[6] P. Angelini, F. Frati, M. Patrignani, and V. Roselli, *Morphing planar graph drawings efficiently*, in Proceedings of the 21st International Symposium on Graph Drawing (GD '13), S. Wismath and A. Wolff, eds., Lecture Notes in Comput. Sci. 8242, Springer, Cham, Switzerland, 2013, pp. 49–60.

[7] B. Aronov, R. Seidel, and D. L. Souvaine, *On compatible triangulations of simple polygons*, Comput. Geom., 3 (1993), pp. 27–35.

[8] G. Barequet and M. Sharir, *Piecewise-linear interpolation between polygonal slices*, Comput. Vis. Image Understanding, 63 (1996), pp. 251–272.

[9] F. Barrera-Cruz, P. Haxell, and A. Lubiw, *Morphing planar graph drawings with unidirectional moves*, in Mexican Conference on Discrete Mathematics and Computational Geometry, Oaxaca, Mexico, 2013, pp. 57–65.

[10] F. Barrera-Cruz, P. E. Haxell, and A. Lubiw, *Morphing Schnyder drawings of planar triangulations*, in Proceedings of the 22nd International Symposium on Graph Drawing (GD '14), C. A. Duncan and A. Symvonis, eds., Lecture Notes in Comput. Sci. 8871, Springer, Heidelberg, 2014, pp. 294–305.

[11] T. Beier and S. Neely, *Feature-based image metamorphosis*, in Proceedings of the 19th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '92), J. J. Thomas, ed., ACM, New York, 1992, pp. 35–42.

[12] T. Biedl, A. Lubiw, M. Petrick, and M. Spriggs, *Morphing orthogonal planar graph drawings*, ACM Trans. Algorithms, 9 (2013), 29.

[13] S. Cairns, *Deformations of plane rectilinear complexes*, Amer. Math. Monthly, 51 (1944), pp. 247–252.

[14] E. Carmel and D. Cohen-Or, *Warp-guided object-space morphing*, Vis. Comp., 13 (1997), pp. 465–478.

[15] N. Chiba, T. Yamanouchi, and T. Nishizeki, *Linear algorithms for convex drawings of planar graphs*, in Progress in Graph Theory, J. A. Bondy and U. S. R. Murty, eds., Academic Press, New York, 1984, pp. 153–173.

[16] R. Connelly, E. D. Demaine, and G. Rote, *Straightening polygonal arcs and convexifying polygonal cycles*, Discrete Comput. Geom., 30 (2003), pp. 205–239.

[17] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, *Computational Geometry*, Springer, Berlin, 2000.

[18] E. D. Demaine and J. O'Rourke, *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*, Cambridge University Press, New York, 2007.

[19] C. Erten, S. G. Kobourov, and C. Pitta, *Intersection-free morphing of planar graphs*, in Proceedings of the 11th International Symposium on Graph Drawing (GD '03), G. Liotta, ed., Lecture Notes in Comput. Sci. 2912, Springer, Berlin, 2003, pp. 320–331.

[20] C. Erten, S. G. Kobourov, and C. Pitta, *Morphing planar graphs*, in Proceedings of the 20th Annual Symposium on Computational Geometry (SoCG '04), J. Snoeyink and J. Boissonnat, eds., ACM, New York, 2004, pp. 451–452.

[21] H. Everett, S. Lazard, S. M. Robbins, H. Schröder, and S. Whitesides, *Convexifying star-shaped polygons*, in Proceedings of the 10th Canadian Conference on Computational Geometry (CCCG '98), Montreal, Quebec, 1998, pp. 10–12.

[22] M. S. Floater and C. Gotsman, *How to morph tilings injectively*, J. Comput. Appl. Math., 101 (1999), pp. 117–129.

[23] C. Friedrich and P. Eades, *Graph drawing in motion*, J. Graph Algorithms Appl., 6 (2002), pp. 353–370.

[24] K. Fujimura and M. Makarov, *Foldover-free image warping*, Graph. Models Image Process., 60 (1998), pp. 100–111.

[25] C. Gotsman and V. Surazhsky, *Guaranteed intersection-free polygon morphing*, Comput. Graph., 25 (2001), pp. 67–75.

[26] L. Guibas and J. Hershberger, *Morphing simple polygons*, in Proceedings of the 10th Annual Symposium on Computational Geometry (SoCG '94), ACM, New York, 1994, pp. 267–276.

[27] S. H. Hong and H. Nagamochi, *Convex drawings of hierarchical planar graphs and clustered planar graphs*, J. Discrete Algorithms, 8 (2010), pp. 282–295.

[28] S. G. Kobourov and M. Landis, *Morphing planar graphs in spherical space*, J. Graph Algorithms Appl., 12 (2008), pp. 113–127.

[29] A. Lubiw, *Morphing planar graph drawings*, in Canadian Conference on Computational Geometry (CCCG '07), Ottawa, Ontario, 2007, p. 1.

[30] A. Lubiw and M. Petrick, *Morphing planar graph drawings with bent edges*, J. Graph Algorithms Appl., 15 (2011), pp. 205–207.

[31] A. Lubiw, M. Petrick, and M. J. Spriggs, *Morphing orthogonal planar graph drawings*, in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '06), ACM, New York, 2006, pp. 222–230.

[32] M. Nöllenburg, D. Merrick, A. Wolff, and M. Benkert, *Morphing polylines: A step towards continuous generalization*, Comput. Env. Urban Syst., 32 (2008), pp. 248–260.

[33] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer, New York, 1985.

[34] V. Roselli, *Morphing and Visiting Drawings of Graphs*, PhD thesis, Department of Engineering, Roma Tre University, Rome, Italy, 2014.

[35] T. W. Sederberg, P. Gao, G. Wang, and H. Mu, *2-d shape blending: an intrinsic solution to the vertex path problem*, in Proceedings of the 20th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93), ACM, New York, 1993, pp. 15–18.

[36] T. W. Sederberg and E. Greenwood, *A physically based approach to 2-d shape blending*, in Proceedings of the 19th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '92), J. J. Thomas, ed., ACM, New York, 1992, pp. 25–34.

[37] M. Shapira and A. Rappaport, *Shape blending using the star-skeleton representation*, IEEE Comput. Graph. Appl., 15 (1995), pp. 44–50.

[38] V. Surazhsky, *Morphing planar triangulations*, master's thesis, Technion - Israel Institute of Technology, Haifa, Israel, 1999.

[39] V. Surazhsky and C. Gotsman, *Controllable morphing of compatible planar triangulations*, ACM Trans. Graphics, 20 (2001), pp. 203–231.

[40] V. Surazhsky and C. Gotsman, *Intrinsic morphing of compatible triangulations*, Int. J. Shape Model., 9 (2003), pp. 191–201.

[41] A. Tal and G. Elber, *Image morphing with feature preserving texture*, Comput. Graphics Forum, 18 (1999), pp. 339–348.

[42] C. Thomassen, *Deformations of plane graphs*, J. Combin. Theory Ser. B, 34 (1983), pp. 244–257.

[43] C. Thomassen, *Plane representations of graphs*, in Progress in Graph Theory, J. A. Bondy and U. S. R. Murty, eds., Academic Press, New York, 1984, pp. 43–69.

[44] W. T. Tutte, *How to Draw a Graph*, Proc. Lond. Math. Soc. (3), 13 (1963), pp. 743–767.