# Morphing Planar Graph Drawings with a Polynomial Number of Steps

Soroush Alamdari* Patrizio Angelini† Timothy M. Chan* Giuseppe Di Battista†
Fabrizio Frati‡ Anna Lubiw* Maurizio Patrignani† Vincenzo Roselli† Sahil Singla*
Bryan T. Wilkinson§

**Abstract**

In 1944, Cairns proved the following theorem: given any two straight-line planar drawings of a triangulation with the same outer face, there exists a morph (i.e., a continuous transformation) between the two drawings so that the drawing remains straight-line planar at all times. Cairns's original proof required exponentially many morphing steps. We prove that there is a morph that consists of $O(n^2)$ steps, where each step is a linear morph that moves each vertex at constant speed along a straight line. Using a known result on compatible triangulations this implies that for a general planar graph $G$ and any two straight-line planar drawings of $G$ with the same embedding, there is a morph between the two drawings that preserves straight-line planarity and consists of $O(n^4)$ steps.

## 1 Introduction

A morph between two geometric shapes is a continuous transformation of one shape into the other. Computer Graphics, Animation, and Modeling are only few of the areas of Computer Science that make use of morphs. The usual goal in morphing is to ensure that the structure of the shapes be "visible" throughout the entire transformation.

Two-dimensional graph drawings can be used to represent many of the shapes for which morphs are of interest (e.g., two-dimensional images [6, 17, 32], polygons and poly-lines [1, 2, 9, 14, 19, 24, 25, 26, 27]). As such, morphs of graph drawings have been well studied.

In this context, the input consists of two drawings $\Gamma_1$ and $\Gamma_2$ of the same graph $G$, and the problem is to transform continuously from one drawing to the other. A *morph* between $\Gamma_1$ and $\Gamma_2$ 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 $\Gamma_1$ and the drawing at time $t = 1$ is $\Gamma_2$. Preserving 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 [10, 11].

**Morphing Planar Graph Drawings.** This paper is about morphing planar graphs. A result by Angelini et al. [3] addresses the problem of "topological morphing" where the planar embedding changes. However most work, including ours, is about the case where the initial and final planar drawings are *topologically equivalent*—i.e., have the same faces and the same outer face—and planarity must be preserved throughout the morph.

In addition to the above applications, morphing while preserving planarity has application to the problem of creating three-dimensional models from two-dimensional slices [5], with time playing the role of the third dimension. The paper by Lubiw and Petrick [22] bounds the complexity of a morph that preserves planarity but relaxes the straight-line condition to allow edges to be drawn as polylines. The case of orthogonal graph drawings is well-solved—a SODA'06 paper describes an efficiently computable morph between any two orthogonal drawings of the same graph that preserves planarity and orthogonality [23]. The case of preserving more general edge directions is explored by Biedl et al. [7] and in the thesis of Spriggs [28].

**Morphing Straight-Line Planar Drawings.** In this paper we give an efficient algorithm to morph between two topologically equivalent straight-line planar drawings of a triangulated graph, where the morph must preserve straight-line planarity. The issue is to find the vertex trajectories.

**Existence.** In 1944, Cairns [8] gave a proof of existence using an induction argument where at each step a low-degree vertex is contracted to a neighbour in the source drawing. Because the same contraction may not preserve planarity in the target drawing, Cairns needed an extra morphing step in which a graph that is triangulated except for one non-convex face is morphed to make that face convex. More details can be found in Section 2. Since Cairns used two recursive calls

*David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada, {s26hosseinialamdari, tmchan, alubiw, s2singla}@uwaterloo.ca

†Dipartimento di Informatica e Automazione, Roma Tre University, Italy, {angelini, gdb, patrigna, roselli}@dia.uniroma3.it

‡School of Information Technologies, The University of Sydney, Australia, brillo@it.usyd.edu.au

§MADALGO, Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation, Department of Computer Science, Aarhus University, Denmark, btw@cs.au.dk

his method takes an exponential number of steps.

Thomassen [33] extended Cairns's algorithm to all planar straight-line drawings. By augmenting both drawings to isomorphic ("compatible") triangulations he reduced the general case to Cairns's result. The idea of compatible triangulations was rediscovered and thoroughly explored by Aronov et al. [4], 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.

We note that Thomassen [33] also showed that there exists a morph that preserves convexity in addition to straight-line planarity.

**Continuous Morph.** An alternative approach to morphing planar straight-line triangulations was presented by Floater and Gotsman [15]. Their method is based on Tutte's graph drawing algorithm [34], in which the position of each vertex is expressed as a convex combination of the positions of its neighbours. Any matrix of such coefficients provides a planar drawing. So Floater and Gotsman expressed each of the given drawings by means of a matrix of coefficients, and then took a linear interpolation between the matrices to obtain the morph. Gotsman and Surazhsky [18, 29, 30, 31] extended the method to all planar straight-line graph drawings, and showed that the resulting morphs are visually appealing. The algorithm does not produce explicit vertex trajectories. It computes, at any given time-point of the morph, a "snapshot" of the graph at that time. In other words, it provides an efficient implementation of a black-box representation of the morph, where an input time-point is mapped to an output graph drawing. There are no quality guarantees about the number of time-points required to approximate continuous motion, or about the grid size of the drawings or the minimum feature size (the measure of how close a vertex and non-incident edge may be). For related results, see [12, 13, 16].

**Piece-Wise Linear Morphs.** The problem of finding a planar straight-line morph that uses a polynomial number of discrete steps has been asked several times (see, e.g., [20, 21, 22, 23]). The most natural definition of a discrete step is a *linear morph*, where 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 $\Gamma_1$ and $\Gamma_2$ are straight-line planar drawings of a graph, we use $\langle \Gamma_1, \Gamma_2 \rangle$ to denote the linear morph from $\Gamma_1$ to $\Gamma_2$. We seek a morph that consists of a sequence of $k$ linear morphs (a "piece-wise linear morph"). Such a morph can be specified by $k+1$ planar straight-line graph drawings. If $\Gamma_1, \ldots, \Gamma_{k+1}$ are straight-line planar drawings of a graph, we use $\langle \Gamma_1, \ldots, \Gamma_{k+1} \rangle$ to denote the morph from $\Gamma_1$ to $\Gamma_{k+1}$ that consists of the $k$ linear morphs from $\Gamma_i$ to $\Gamma_{i+1}$ for $i = 1, \ldots, k$.

**Our Result.** We solve the above open problem by giving an efficient version of Cairns's algorithm. Given a triangulation $G$ on $n$ vertices and two straight-line planar drawings of $G$ with the same outer face, we give a morph between the two drawings that preserves straight-line planarity and consists of $O(n^2)$ linear morphs. By the result of Aronov, et al. on compatible triangulations [4], it immediately follows that we can morph between two straight-line planar drawings of any planar graph with a common planar embedding using $O((n^2)^2) = O(n^4)$ linear morphing steps.

Our idea is to follow Cairns's vertex contraction proof, but avoid the double recursion by directly giving a morph to convexify one small non-convex face in an otherwise triangulated graph. The other issue is that contracting a vertex to a neighbour is not actually legitimate in a morph. We show how to keep vertices close but not coincident without increasing the number of morphing steps. We begin with a more detailed description of Cairns's approach in the following Section 2. In Sections 3 and 4 we give an efficient "pseudo-morph" that permits vertex contractions, and in Section 5 we show how to avoid coincident vertices.

## 2 Overview of Cairns's Proof

We are given two drawings $\Gamma_1$ and $\Gamma_2$ of a triangulated graph $G$ with the same outer face. Cairns assumed that the outer face is drawn as the same triangle in both drawings, which is easy to accomplish via a few linear morphs. By the identity $\sum_p \deg(p) = 6n - 12$, there is at least one non-boundary vertex $p$ with $\deg(p) \leq 5$. Let $\Delta_i(p)$ be the polygon defined by the neighbourhood of this vertex $p$ in $\Gamma_i$ and let $\Delta_i^*(p)$ be the *kernel* of $\Delta_i(p)$, i.e., the region inside $\Delta_i(p)$ that is visible to all points in $\Delta_i(p)$. We will write $\Gamma$, $\Delta(p)$, and $\Delta^*(p)$ without the subscript $i$ when referring to the drawing at the "current" time in the morph.

Cairns observed the following fact, which follows from a straightforward case analysis: for any ($\leq 5$)-gon, at least one of its vertices must be in the kernel. So, $\Delta_1^*(p)$ contains at least one vertex $t_1$ of $\Delta_1(p)$, and $\Delta_2^*(p)$ contains at least one vertex $t_2$ of $\Delta_2(p)$. It may not be the case that $t_1 = t_2$ (and there may be no vertex that is in both kernels $\Delta_1^*(p)$ and $\Delta_2^*(p)$). However, Cairns [8] observed that we can find a re-drawing $\Gamma_3$ of $G$ in which both $t_1$ and $t_2$ are in the kernel $\Delta_3^*(p)$. The precise details are not important here, but the general idea is to find a re-drawing $\Gamma_3$ in which the polygon $\Delta_3(p)$ is convex, and if we are unable to convexify the polygon due to the presence of *external chords* (i.e., edges between two vertices of $\Delta(p)$ that lie outside $\Delta(p)$), the idea is to make the polygon $\Delta_3(p)$ "as convex as possible".

Once we have found this intermediate drawing $\Gamma_3$, we can find a morph from $\Gamma_1$ to $\Gamma_2$ as follows:

1. Morph from $\Gamma_1$ to $\Gamma_3$: To accomplish this, we first contract $p$ to $t_1$ in both drawings $\Gamma_1$ and $\Gamma_3$. The contraction can be done by a linear morph without

violating planarity because $t_1$ is in both kernels $\Delta_1^*(p)$ and $\Delta_3^*(p)$. We can then use recursion to morph between the two drawings with $n-1$ vertices.

2. Morph from $\Gamma_3$ to $\Gamma_2$: To accomplish this, we first contract $p$ to $t_2$ in both drawings $\Gamma_2$ and $\Gamma_3$. The contraction can again be done without violating planarity because $t_2$ is in both kernels $\Delta_2^*(p)$ and $\Delta_3^*(p)$. We can again use recursion to morph between the two drawings with $n-1$ vertices.

The number of linear morphing steps required by the resulting algorithm satisfies the recurrence

$$T(n) \le 2T(n-1) + O(1), \tag{2.1}$$

yielding $T(n) = O(2^n)$, which is exponential, unfortunately.

There is another problem with the above description: we cannot move $p$ to lie exactly on top of $t_1$ or $t_2$ during the morph, since "coincident" vertices are technically not allowed in a drawing. What the above algorithm finds is a *pseudo-morph*, which we define inductively as a sequence consisting of the following kinds of steps:

- a linear morph;
- a contraction of a vertex $p$ to another vertex, followed by a pseudo-morph between the two reduced drawings, and then an "un-contraction" of $p$.

In Cairns's original proof, after computing the morph between the two reduced drawings, we add $p$ back to the morph by setting $p$ to lie somewhere inside the kernel $\Delta^*(p)$ at all times during the morph. Specifically, Cairns suggested placing $p$ at the centroid of $\Delta^*(p)$, which is fine for an existential proof, but introduces further complexity. Appendix A contains an example where the the centroid of $\Delta^*(p)$ does not move in a straight line during a linear morph of the drawing. We can keep $p$ inside the kernel $\Delta^*(p)$ by breaking its trajectory into linear pieces and using a sequence of linear morphs, but because the kernel may experience many combinatorial changes over time, there is no obvious way to bound the number of linear morphs. The end result may be more exponential factors if the coefficient in front of $T(n-1)$ in (2.1) is increased.

For now, we will ignore the issue of how to turn a pseudo-morph into an actual morph that avoids coincident vertices; we will return to this issue later in Section 5. In Sections 3–4, we first concentrate on developing an algorithm that can find a pseudo-morph with $O(n^2)$ steps.

## 3 New Approach

To improve Cairns's algorithm, the first main idea is to bypass the first of the two recursive calls. Instead of morphing from $\Gamma_1$ to a fixed target drawing $\Gamma_3$, the subproblem we actually need to solve is a weaker one:

PROBLEM 3.1. ($(\le 5)$-GON "CONVEXIFICATION") *Given a vertex $p$ of degree at most 5 with $t_1$ in $\Delta_1^*(p)$ and $t_2$ in $\Delta_2^*(p)$, find a pseudo-morph from $\Gamma_1$ to* any *straight-line planar drawing $\Gamma_3$ of $G$ in which $t_2$ is in $\Delta_3^*(p)$.*

We put "convexification" in quotes, since we do not require that $\Delta_3(p)$ is completely convex, just that its kernel contains the specified vertex $t_2$. Let $T_{\text{conv5}}(n)$ denote the minimum number of linear morphing and contraction/uncontraction steps required to solve Problem 3.1. Then the recurrence for the overall morphing algorithm changes from (2.1) to

$$T(n) \le T(n-1) + T_{\text{conv5}}(n) + O(1), \tag{3.2}$$

which is polynomial if $T_{\text{conv5}}(n)$ is polynomial.

To solve the $(\le 5)$-gon "convexification" problem, it is natural to start with the 4-gon convexification problem: Given a vertex $p$ of degree 4 with $t_1$ in $\Delta_1^*(p)$ and $t_2$ in $\Delta_2^*(p)$, find a pseudo-morph from $\Gamma_1$ to *any* straight-line planar drawing $\Gamma_3$ of $G$ in which $t_2$ is in $\Delta_3^*(p)$. Observe that in a non-convex 4-gon $abcd$, say with $d$ as the reflex vertex as in Figure 3, both $d$ and the opposite vertex $b$ are in the kernel. Thus the only case we need to convexify is when the vertex $t_2$ is $a$ or $c$. Because $t_2$ is in $\Delta_2^*(p)$, the edge $ac$ must be inside the 4-gon in $\Gamma_2$, and therefore cannot be outside the 4-gon in $\Gamma_1$. Thus the 4-gon convexification problem can be rephrased as follows:

PROBLEM 3.2. (4-GON CONVEXIFICATION) *Given a triangulated graph $G$ with a triangle boundary and a 4-gon $abcd$ in a straight-line planar drawing of $G$ such that neither $ac$ nor $bd$ is an edge outside of $abcd$ (i.e., $abcd$ does not have external chords), find a pseudo-morph so that $abcd$ becomes convex.*

Let $T_{\text{conv4}}(n)$ denote the minimum number of linear morphing and contraction/uncontraction steps required to solve Problem 3.2. We will solve Problem 3.2 in the next section. Our idea is to adapt Cairns's approach again. Namely, we find a low-degree vertex, contract it to a neighbor, and recursively solve the 4-gon convexification problem on a graph with $n-1$ vertices. No second recursive call is required this time. The details involve a case analysis, and the end result is the recurrence

$$T_{\text{conv4}}(n) \le T_{\text{conv4}}(n-1) + O(1), \tag{3.3}$$

yielding a linear bound $T_{\text{conv4}}(n) = O(n)$.

One might think that 5-gon convexification would require an even longer case analysis, but we show that the 5-gon case can be reduced to the 4-gon case:

LEMMA 3.1. $T_{\text{conv5}}(n) = O(T_{\text{conv4}}(n-1))$.

*Proof.* We solve Problem 3.1 using a black-box solution to Problem 3.2. We initially contract $p$ to $t_1$ in $\Gamma_1$ (recall that $t_1$ is in the kernel $\Delta_1^*(p)$). If $p$ has degree 3, then we are done, and if $p$ has degree 4, then we can convexify $\Delta(p)$ using at most $T_{\text{conv4}}(n-1)$ steps. So assume that $\Delta(p)$ is a 5-gon, of the form $t_2abcd$.

First consider the case where $t_1$ is $b$. Thus $p$ is contracted to $b$ which is in the kernel of the polygon (see Figure 1). We convexify the 4-gon $t_2bcd$ using at most $T_{\text{conv4}}(n-1)$ steps. The convexification is possible since $t_2c$ cannot be an edge outside $t_2bcd$, for otherwise $t_2$ cannot be in the kernel $\Delta_2^*(p)$. Now, $t_2$ sees all of the 4-gon $t_2bcd$ as well as the triangle $t_2ab$ and thus lies in the kernel $\Delta^*(p)$. At the end we uncontract $p$.

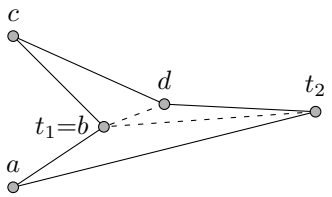



Figure 1: Contraction of $p$ to $b$

Next consider the case where $t_1$ is $d$. Thus $p$ is contracted to $d$ which is in the kernel of the polygon (see Figure 2). We convexify the 4-gon $t_2abd$ using at most $T_{\text{conv4}}(n-1)$ steps. The convexification is possible since $t_2b$ cannot be an edge outside $t_2abd$, for otherwise $t_2$ cannot be in the kernel $\Delta_2^*(p)$. Now, $b$ sees all of the 4-gon $t_2abd$ as well as the triangle $bcd$ and thus lies in the kernel $\Delta^*(p)$. We uncontract $p$ from $d$, contract $p$ to $b$, and arrive at the previous case.

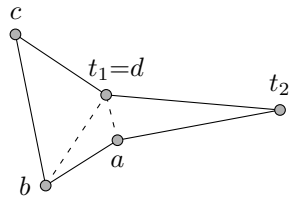



Figure 2: Contraction of $p$ to $d$

The remaining cases are symmetric.

Combining (3.2) and (3.3) with Lemma 3.1 gives the recurrence $T(n) \leq T(n-1) + O(n)$, implying that $T(n) = O(n^2)$. All that remains is to solve the 4-gon convexification problem and prove (3.3).

## 4 Convexifying a 4-Gon

In this section we present a solution to Problem 3.2. Assume we are given a straight-line planar drawing $\Gamma$ of a triangulated graph with a triangle boundary $z_1z_2z_3$, and we are given a 4-gon $abcd$ such that neither $ac$ nor $bd$ is an edge outside of $abcd$. We want to find a pseudo-morph that convexifies $abcd$ while preserving straight-line planarity throughout. The boundary $z_1z_2z_3$ stays fixed during the morph. If $abcd$ is convex, we are done. Assume without loss of generality that $d$ is the reflex vertex of the non-convex 4-gon, and thus $b$ is the tip of the arrowhead shape (see Figure 3). The triangulation contains the edge $bd$.

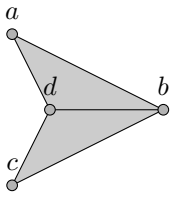



Figure 3: 4-Gon $abcd$

We follow Cairns's idea of reducing the size of the problem by one by contracting a non-boundary vertex $p$ with $\deg(p) \leq 5$ to a vertex that is in $\Delta^*(p)$ and then recursively finding a pseudo-morph. However, we need to be careful not to destroy $abcd$ or introduce the edge $ac$ when we contract $p$, since the external chord $ac$ would preclude convexification of $abcd$. We supplement Cairns's approach with additional arguments to deal with different types of problematic vertices $p$.

Formally, we define a *problematic* vertex to be a vertex $p$ of one of the following types:

- $p$ is a vertex of the 4-gon $abcd$ and is not on the boundary.
- $p$ is a vertex of the boundary triangle $z_1z_2z_3$.
- $p$ is outside the 4-gon, is not on the boundary, has degree at most 5, and is adjacent to both $a$ and $c$, and either $a$ or $c$ is in $\Delta^*(p)$.

We call a vertex $p$ of the third type an *ac-inducing* vertex, since following Cairns's original approach and attempting to contract $p$ to $a$ or $c$ would introduce the forbidden edge $ac$.

Define the *value* of a vertex $p$ to be $6 - \deg(p)$. The total value of all vertices is $6n - \sum_p \deg(p) = 12$.

If there is a non-problematic vertex $p$ with $\deg(p) \leq 5$, then Cairns's approach works fine. Thus, we may assume that every non-problematic vertex has degree at least 6, i.e., value at most 0. In the next three subsections, we show that some kinds of problematic vertices can be handled, that is, we can either morph directly to convexify the 4-gon $abcd$,

or we can perform a contraction that reduces the problem size by one. When no problematic vertices can be handled, we bound the values of the remaining problematic vertices. Then in Section 4.4 we rule out the bad case where no vertex can be handled using a counting argument based on the vertex values.

**4.1 Non-Boundary $4$-Gon Vertices** Suppose that $a$ is not a boundary vertex and $\deg(a) \leq 4$ (see Figure 4). Observe the following fact: in any $(\leq 4)$-gon, every edge must be incident to at least one vertex that lies in the kernel (this is obvious by picturing the case of a non-convex 4-gon). In particular, $b$ or $d$ must be in $\Delta^*(a)$. We can thus move $a$ to within a small distance from whichever vertex is in $\Delta^*(a)$ and perturb it slightly to directly convexify $abcd$.

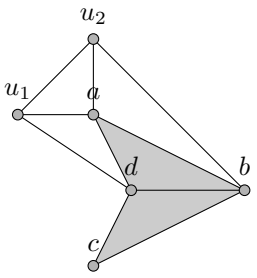



Figure 4: Non-boundary vertex $a$

The case of a degree-$(\leq 4)$ non-boundary vertex $c$ is symmetric.

Suppose that $b$ is not a boundary vertex and $\deg(b) \leq 4$ (see Figure 5). Then $b$ must have exactly one neighbour $x$ that is not on $abcd$, and $x$ must be adjacent to both $a$ and $c$. Since $d$ is a reflex vertex of $axcd$, $x$ must be in $\Delta^*(b)$. So, we contract $b$ to $x$, recursively convexify $axcd$, and uncontract $b$.

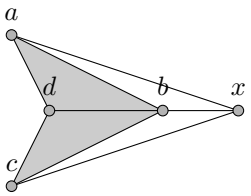
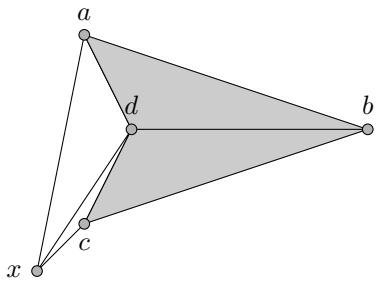
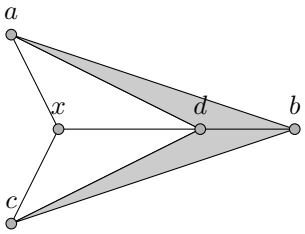



Figure 5: Non-boundary vertex $b$

Suppose that $d$ is not a boundary vertex and $\deg(d) \leq 4$ (see Figure 6). Then $d$ must have exactly one neighbour $x$ that is not on $abcd$, and $x$ must be adjacent to both $a$ and $c$. If $x$ is outside of the triangle $abc$, then $a$ is in $\Delta^*(d)$, and we directly convexify $abcd$ by moving $d$ to within a small distance from $a$. Otherwise, we contract $d$ to $x$, recursively convexify $abcx$, and uncontract $d$.




(a) $x$ outside triangle $abc$




(b) $x$ inside triangle $abc$

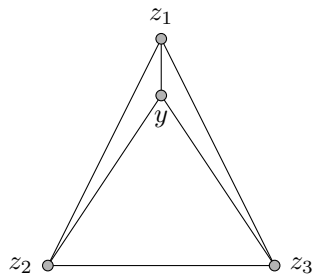
Figure 6: Non-boundary vertex $d$

We may now assume that each non-boundary vertex of $abcd$ has degree at least 5, i.e., value at most 1, for a total value of at most 4.

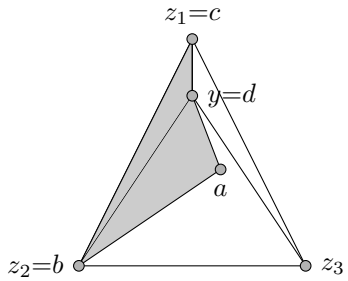
**4.2 Boundary Vertices** Suppose a boundary vertex, say, $z_1$, has degree 3. Then $z_1$ must have a neighbour $y$ that is also adjacent to $z_2$ and $z_3$ (see Figure 7a). Note that the triangles $z_1yz_2$ and $z_1yz_3$ are empty of other vertices.

If $abcd$ lies entirely within the triangle $T = yz_2z_3$, then we can simply recursively morph the subgraph contained in $T$. The fact that edge $ac$ does not exist limits any other cases we must consider. In particular, $abcd$ must be composed of either $z_1yz_2$ or $z_1yz_3$ and an adjacent triangle in $yz_2z_3$. Without loss of generality, assume the former (see Figure 7b). Then, we move $y$ to within a small distance from $z_1$ and perturb it slightly to directly convexify $abcd$. As we move $y$, we linearly transform the contents of $yz_2z_3$.

Now, suppose that all three boundary vertices have degree 4. Then there must exist an internal triangle $y_1y_2y_3$ containing all the vertices except for the boundary vertices with $z_i$ adjacent to $y_j$ for $j \neq i$, $i, j \in \{1, 2, 3\}$. If $abcd$ lies entirely within $y_1y_2y_3$ (see Figure 8a), then we simply recursively morph the subgraph contained in $y_1y_2y_3$. If $abcd$ lies entirely outside $y_1y_2y_3$, then consider the graph of constant size formed by deleting all vertices within $y_1y_2y_3$. It can be verified that the polygon defined by any pair of adjacent faces in this graph can be made convex in a constant number of steps. Thus, if $abcd$ is composed of two of the faces of this graph, then we are done. The final case occurs

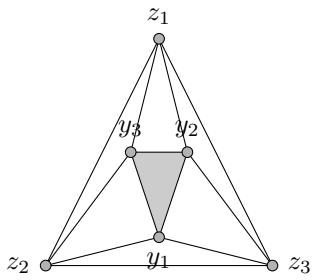



(a)

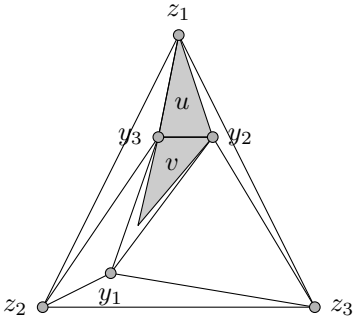



(b)

Figure 7: Boundary vertex with degree 3




(a) $abcd$ within triangle $y_1y_2y_3$




(b) $abcd$ composed of triangles $u$ and $v$

Figure 8: All three boundary vertices with degree 4

when $abcd$ is composed of a triangle $u$ outside $y_1y_2y_3$ and a triangle $v$ inside $y_1y_2y_3$ (see Figure 8b). We convexify the polygon defined by $u$ and $y_1y_2y_3$ and linearly transform the contents of $y_1y_2y_3$ throughout the morph. This morph necessarily also convexifies the polygon defined by $u$ and $v$, which is $abcd$.

We may now assume that the boundary vertices have total degree at least $4 + 4 + 5$, i.e., total value at most $2 + 2 + 1 = 5$.

**4.3 $ac$-Inducing Vertices** Recall that a non-boundary vertex $x$ is $ac$-inducing if $\deg(x) \leq 5$, $x$ is adjacent to both $a$ and $c$, and either $a$ or $c$ is in $\Delta^*(x)$ (see Figure 9a).

Our key observation is that such vertices are rare:

LEMMA 4.1. *There are at most two $ac$-inducing vertices.*

*Proof.* Assume towards contradiction that there are two $ac$-inducing vertices, $x$ and $x'$ on the same side of $\overline{ac}$. Since either $a$ or $c$ is in $\Delta^*(x)$, $\overline{ac}$ lies inside $\Delta(x)$. However, then $\overline{ac}$ can not also lie inside $\Delta(x')$, so we have a contradiction. Therefore, there are at most two $ac$-inducing vertices: at most one on each side of $\overline{ac}$.

We first note that no $ac$-inducing vertex has degree 3; otherwise, $ac$ would necessarily be an edge, which is a contradiction.

Now, suppose that there are two $ac$-inducing vertices, $x$ and $x'$, where $\deg(x) = 4$ and $\deg(x') \in \{4, 5\}$ (see Figure 9b). Polygon $axcx'$ cannot contain any vertices, or else $a$ and $c$ would not be able to see each other in either $\Delta(x)$ or $\Delta(x')$, and thus one of $x$ or $x'$ would not be an $ac$-inducing vertex.

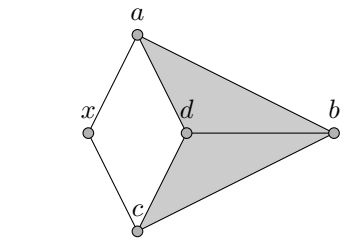
Assume without loss of generality that $a$ is in $\Delta^*(x')$. We first move $x'$ to within a small distance from $a$ and outside $abc$. Inside $abc$, $\Delta(x)$ is a triangle, and so $x'$ lies in $\Delta^*(x)$. We contract $x$ to $x'$, recursively convexify in the reduced graph, and uncontract $x$.

We may now assume either that there is exactly one $ac$-inducing vertex, with degree at least 4, or that there are exactly two $ac$-inducing vertices, with degrees at least $4 + 6$ or $5 + 5$. In any case, the $ac$-inducing vertices have total value at most 2.

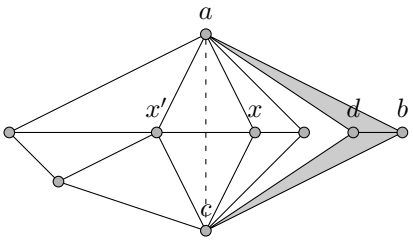
**4.4 Putting It All Together** To summarize, if none of the preceding cases is applicable, then

- the non-problematic vertices have total value at most 0,
- the non-boundary vertices of $abcd$ have total value at most 4,
- the boundary vertices have total value at most 5, and
- the $ac$-inducing vertices have total value at most 2.

(It is possible to eliminate at least one more case with further arguments, but this would not be necessary.) Since the total




(a) $ac$-inducing vertex, $x$




(b) Two $ac$-inducing vertices, $x$ and $x'$

Figure 9: $ac$-inducing vertices

value of all vertices is equal to 12, we have already reached a contradiction. We conclude that there is a 4-gon convexification algorithm satisfying the recurrence (3.3), yielding a pseudo-morph with $O(n)$ steps. As a consequence, there is a pseudo-morph with $O(n^2)$ steps between any two straight-line planar drawings of a triangulated graph with a common triangle boundary.

## 5 Avoiding Coincident Vertices

The preceding algorithm only constructs pseudo-morphs. In this section, we describe a general way of converting any given pseudo-morph $\mathcal{M}$ into an actual morph $M$, under the assumption that in the pseudo-morph, we only contract non-boundary vertices with degree at most 5. This assumption clearly holds for the algorithm in Sections 3–4.

Suppose that the given pseudo-morph $\mathcal{M}$ consists of the contraction of a non-boundary vertex $p$ with $\deg(p) \leq 5$ to a vertex $a$, followed by a pseudo-morph $\mathcal{M}_0$ of the reduced graph, and then the uncontraction of $p$. We recursively convert $\mathcal{M}_0$ into a morph $M_0$. Then we modify $M_0$ to $M_0^p$ by adding $p$ back to the morph, placing it somewhere inside the kernel $\Delta^*(p)$ at all times, to preserve planarity when we add back the edges incident to $p$. In fact, we will keep $p$ close to $a$, which we know lies in the kernel. To obtain the final morph $M$, we replace the contraction of $p$ to $a$ by a linear morph that moves $p$ from its initial position to its position at the start of $M_0^p$, and we replace the uncontraction of $p$ by a linear morph that moves $p$ from its position at the end of $M_0^p$ to its final position.

Cairns's original proof converts $M_0$ to $M_0^p$ by placing $p$ at the centroid of the kernel, but the overall number of linear morphing steps could increase drastically. We will use a different placement of $p$ that preserves the number of steps exactly.

As a warm-up case, and because we will need it later on, we first consider the case when $\deg(p) = 3$. For this case, it suffices to show that we can add a new point $p$ inside a triangle as it undergoes a linear morph:

LEMMA 5.1. *Let $\Gamma_1$ and $\Gamma_2$ be straight-line drawings of a triangle on vertices $a, b, c$ in clockwise order such that the morph $\langle \Gamma_1, \Gamma_2 \rangle$ is planar. Augment each $\Gamma_i$ to $\Gamma_i^p$ by adding a point $p$ at a convex combination $\lambda_1 a_i + \lambda_2 b_i + \lambda_3 c_i$ for some constants $\lambda_1, \lambda_2, \lambda_3 > 0$ with $\lambda_1 + \lambda_2 + \lambda_3 = 1$, where $a_i$ is the position of $a$ in $\Gamma_i$, etc. Then the morph $\langle \Gamma_1^p, \Gamma_2^p \rangle$ is planar, and furthermore, at each time instant $t$ of the morph, with vertices $p, a, b, c$ at points $p_t, a_t, b_t, c_t$, we have $p_t = \lambda_1 a_t + \lambda_2 b_t + \lambda_3 c_t$.*

Note that for the case when $\deg(p) = 3$ it suffices to consider a single linear morph, because the rule for placing $p$ is uniform. The same is true of the case when $\deg(p) = 4$, and the argument is equally simple: if a 4-gon $a, b, c, d$ undergoes a planar linear morph that keeps $a$ in the kernel, then the line segment $ac$ also remains in the kernel, so we can place $p$ at a fixed linear combination of $a$ and $c$.

We now turn to the general case when $\deg(p) = 5$. In this case we must consider the whole sequence of linear morphs, and the argument is more complicated.

LEMMA 5.2. *Let $\Gamma_1, \ldots, \Gamma_k$ be straight-line planar drawings of a 5-gon $C$ on vertices $a, b, c, d, e$ in clockwise order such that the morph $\langle \Gamma_1, \ldots, \Gamma_k \rangle$ is planar and vertex $a$ is inside the kernel of the polygon $C$ at all times during the morph. Then we can augment each drawing $\Gamma_i$ to a drawing $\Gamma_i^p$ by adding vertex $p$ at some point $p_i$ inside the kernel of the polygon $C$ in $\Gamma_i$, and adding straight line edges from $p$ to each of $a, b, c, d, e$ in such a way that the morph $\langle \Gamma_1^p, \ldots, \Gamma_k^p \rangle$ is planar.*

*Proof.* We assume that vertex $a$ has the same position during the entire morph $\langle \Gamma_1, \ldots, \Gamma_k \rangle$. This is not a loss of generality because if vertex $a$ moves, we can translate the whole drawing to move it back: if $\Gamma_1$ and $\Gamma_2$ are planar straight-line drawings of a graph and $\Gamma_3$ is a translation of $\Gamma_2$. Then $\langle \Gamma_1, \Gamma_2 \rangle$ is a planar morph if and only if $\langle \Gamma_1, \Gamma_3 \rangle$ is.

We also assume that $b$, $a$, and $e$ are not collinear in any drawing $\Gamma_i$, as otherwise we can slightly perturb the position of $a$ without affecting the planarity of the morph $\langle \Gamma_1, \ldots, \Gamma_k \rangle$. Hence, in any drawing $\Gamma_i$, the angle $\alpha_i$ incident to $a$ and internal to the polygon $C$ is such that either $\alpha_i < \pi$ or $\alpha_i > \pi$. If $\alpha_i < \pi$, we say that $\Gamma_i$ is *$a$-convex*. Otherwise, it is *$a$-reflex*.

We partition the sequence of drawings $\Gamma_1, \ldots, \Gamma_k$ into maximal subsequences of $a$-convex and $a$-reflex drawings.

Observe that at any time instant $t$ during morph $\langle \Gamma_1, \ldots, \Gamma_k \rangle$ there exists an $\epsilon_t > 0$ such that the intersection between the disk $D$ centered at $a$ with radius $\epsilon_t$ and the kernel of polygon $C$ consists of a non-zero-area sector $S$ of $D$. This is because $a$ is a vertex of the kernel of $C$. Let $\epsilon = \min_t \epsilon_t$ be the minimum of $\epsilon_t$ among all time instants $t$ of the morph.

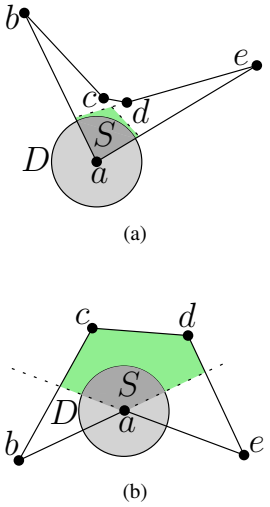



Figure 10: Illustrations for the existence of a disk $D$ (light grey region) centered at $a$ whose intersection with the kernel of $C$ (green region) is a non-zero-area sector $S$ (dark grey region) in (a) the $a$-convex case and (b) the $a$-reflex case.

Observe that if $\Gamma_i$ is $a$-convex then the radii of $D$ delimiting $S$ lie on edges $(a, b)$ and $(a, e)$, respectively (see Fig. 10a), while if $\Gamma_i$ is $a$-reflex then the radii of $D$ delimiting $S$ lie on the elongations of edges $(a, b)$ and $(a, e)$ emanating from $a$, respectively (see Fig. 10b).

In drawing $\Gamma_i$ we denote the position of vertex $w$ by $w_i$. Note that the position of vertex $a$ is unchanged so we abuse the notation and simply denote it by $a$. We denote the length of edge $(w, z)$ in $\Gamma_i$ by $l_i(w, z)$. Let $l(w, z) = \max_i l_i(w, z)$. Define $f_b = \epsilon / l(a, b)$ and $f_e = \epsilon / l(a, e)$.

Consider a maximal subsequence of $\Gamma_1, \ldots \Gamma_k$ that consists entirely of $a$-convex (or entirely of $a$-reflex) drawings. We will construct a triangle $a, b', e'$ with dummy vertices $b'$ and $e'$ so that the triangle lies inside the kernel of the polygon throughout the subsequence. Then the idea is to express vertex $p$ as a fixed convex combination of $a, b', e'$ so that we obtain a planar morph by Lemma 5.1. Dummy vertices $b'$ and $e'$ are placed as follows.

If the subsequence is $a$-convex then for each $\Gamma_i$ in the subsequence, $b'$ and $e'$ are placed at points $b'_i = f_b \cdot b_i + (1 - f_b) \cdot a$ and $e'_i = f_e \cdot e_i + (1 - f_e) \cdot a$, respectively. See Fig. 11a. Observe that $b'$ lies on edge $(a, b)$, since its position is a convex combination of the positions of $a$ and $b$.

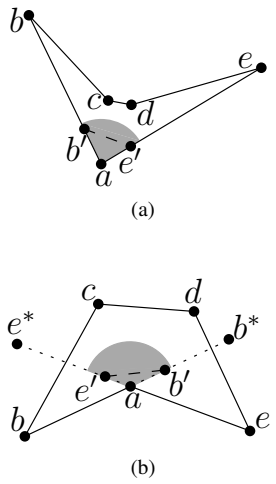



Figure 11: Illustrations for the placement of dummy vertices $b'$ and $e'$ in the $a$-convex and $a$-reflex cases to create triangle $\tau_i$.

Similarly, $e'$ lies on edge $(a, e)$.

If the subsequence is $a$-reflex then for each $\Gamma_i$ in the subsequence, construct point $b_i^*$ by extending the line segment $b_i a$ an equal distance beyond $a$. Similarly, construct point $e_i^*$ by extending the line sement $e_i a$ an equal distance beyond $a$. Vertices $b'$ and $e'$ are placed at points $b'_i = f_b \cdot b_i^* + (1 - f_b) \cdot a$ and $e'_i = f_e \cdot e_i^* + (1 - f_e) \cdot a$, respectively. See Fig. 11b. Observe that $b'$ and $e'$ lie on the elongations of $(a, b)$ and $(a, e)$ emanating from $a$, respectively.

Note also that, whether $\Gamma_i$ is $a$-convex or $a$-reflex, the positive-area triangle $(a, b', e')$ is contained inside the sector $S$ that is the intersection between $D$ and the kernel of $C$. This is because $f_b \cdot l_i(a, b) \leq \epsilon l_i(a, b) / l(a, b) \leq \epsilon$. Similarly $f_e \cdot l_i(a, e) \leq \epsilon$.

We denote by $\tau_i$ the triangle $(a, b', e')$ in drawing $\Gamma_i$.

Observe that, during each morphing step that transforms an $a$-convex drawing $\Gamma_i$ into an $a$-convex drawing $\Gamma_{i+1}$, vertices $b'$ and $e'$ move at constant speed along trajectories that are parallel to those of $b$ and $e$, respectively, since they are expressed as convex combinations of such points and of $a$, which does not move. Hence, they move linearly. Moreover, since $b'$ and $e'$ lie on edges $(a, b)$ and $(a, e)$, respectively, and since the morphing $\langle \Gamma_i, \Gamma_{i+1} \rangle$ is planar, the linear morphing $\langle \tau_i, \tau_{i+1} \rangle$ is also planar.

Analogously, during each morphing step that transforms an $a$-reflex drawing $\Gamma_i$ into an $a$-reflex drawing $\Gamma_{i+1}$, vertices $b'$ and $e'$ move linearly, since they are expressed as convex combinations of $a$ and of $b^*$ and $e^*$, which in turn move parallel to $b$ and $e$, since triangles $(a, b, e)$ and $(a, b^*, e^*)$ are congruent and symmetric with respect to $a$. Moreover, since $b'$ and $e'$ lie on the lines through edges $(a, b)$ and $(a, e)$, respectively, and since the morphing $\langle \Gamma_i, \Gamma_{i+1} \rangle$ is planar, the

linear morphing $\langle \tau_i, \tau_{i+1} \rangle$ is also planar.

On the other hand, during a morphing step from an $a$-convex $\Gamma_i$ to an $a$-reflex $\Gamma_{i+1}$, or vice versa, triangle $\tau_i$ degenerates to a segment and triangle $\tau_{i+1}$ then reappears from the segment. However, it is crucial to note that the intersection between $\tau_i$ and $\tau_{i+1}$ is a non-empty convex region incident to $a$ (see Fig. 12), since it is not possible to move $b$ (resp., $e$) in one single planar linear morph in such a way that its trajectory crosses the elongation of $(a, b)$ (resp., of $(a, e)$) emanating from $a$.

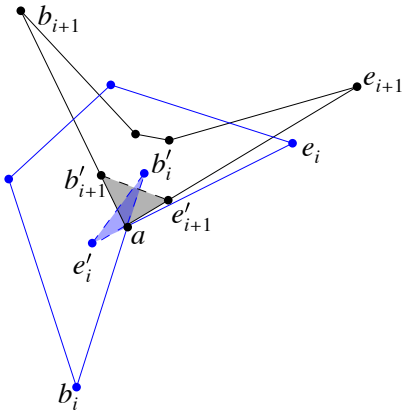



Figure 12: The intersection between $\tau_i$ (blue region) in the $a$-reflex $\Gamma_i$ and $\tau_{i+1}$ (grey region) in the $a$-convex $\Gamma_{i+1}$ is a non-empty region incident to $a$.

Based on the above discussion, we give an algorithm to construct the required morphing $\langle \Gamma_1^p, \ldots, \Gamma_k^p \rangle$.

Our proof is by induction on the number of alternations between $a$-convex and $a$-reflex drawings in the sequence. Assume that $\Gamma_1$ is $a$-reflex (the other case is analogous). In the base case we handle the first maximal subsequence of $a$-reflex drawings. Let $i$ be the minimum index such that $\Gamma_1, \ldots, \Gamma_i$ are $a$-reflex and $\Gamma_{i+1}$ is $a$-convex. For each $r$ with $1 \leq r \leq i$, we will construct $\Gamma_r^p$ from $\Gamma_r$ by placing vertex $p$ at a point $p_r$ and adding straight-line edges from $p$ to vertices of $C$, ensuring that $\langle \Gamma_1^p, \ldots, \Gamma_i^p \rangle$ is a planar morph. Take any point $p_i$ in the intersection of $\tau_i$ and $\tau_{i+1}$. Parameterize $p_i$ as a convex combination of $a$, $b'_i$ and $e'_i$, say $p_i = \lambda_1 a + \lambda_2 b'_i + \lambda_3 e'_i$. Using the same $\lambda$'s, define point $p_r$ to be $\lambda_1 a + \lambda_2 b'_r + \lambda_3 e'_r$. Then, by the above discussion, the morph $\langle \Gamma_1^p, \ldots, \Gamma_i^p \rangle$ is planar. This completes the base case of the induction.

Next, consider a morphing step from an $a$-reflex $\Gamma_i$ to an $a$-convex $\Gamma_{i+1}$. (The step from $a$-convex to $a$-reflex is analogous.) Assume inductively that:

- we have chosen points $p_r$ for $1 \leq r \leq i$ such that the morph $\langle \Gamma_1^p, \ldots, \Gamma_i^p \rangle$ is planar and
- point $p_i$ is inside the region of intersection, $R$, between triangle $\tau_i$ and triangle $\tau_{i+1}$.

We consider the maximal sequence of $a$-convex drawings including $\Gamma_{i+1}$. Let $j > i$ be the minimum index such that $\Gamma_{i+1}, \ldots, \Gamma_j$ are $a$-convex and $\Gamma_{j+1}$ is $a$-reflex. For each $r$ with $i + 1 \leq r \leq j$, we will construct $\Gamma_r^p$ from $\Gamma_r$ by placing vertex $p$ at a point $p_r$ and adding straight-line edges from $p$ to vertices of $C$. Our goal is to ensure that $\langle \Gamma_i^p, \Gamma_{i+1}^p, \ldots, \Gamma_j^p \rangle$ is planar. Our plan is to choose parameters $\lambda_1, \lambda_2, \lambda_3$ and keep vertex $p$ at the point $\lambda_1 a + \lambda_2 b' + \lambda_3 e'$ throughout. This ensures planarity of all steps of the morph except possibly the first step.

We have a constraint at the end of the sequence, that $p_j$ should lie in $\tau_j \cap \tau_{j+1}$. We also have to take care that the first step $\langle \Gamma_i^p, \Gamma_{i+1}^p \rangle$ is planar.

The constraint that $p_j$ should lie in $\tau_j \cap \tau_{j+1}$ restricts the choice of $\lambda_1, \lambda_2, \lambda_3$ to the set $\Lambda$ defined as $\{(\lambda_1, \lambda_2, \lambda_3) \colon \lambda_1, \lambda_2, \lambda_3 > 0, \lambda_1 + \lambda_2 + \lambda_3 = 1, \lambda_1 a + \lambda_2 b'_j + \lambda_3 e'_j \in \tau_j \cap \tau_{j+1}\}$. Define $R'$ to be the region inside $\tau_{i+1}$ corresponding to $\Lambda$, viz. $\{\lambda_1 a + \lambda_2 b'_{i+1} + \lambda_3 e'_{i+1} \colon (\lambda_1, \lambda_2, \lambda_3) \in \Lambda\}$. Refer to Fig. 13a. It remains to choose a point $p_{i+1}$ in $R'$ so that the morph $\langle \Gamma_i^p, \Gamma_{i+1}^p \rangle$ is planar.

Consider the morph $\langle \Gamma_i, \Gamma_{i+1} \rangle$. Parameterizing this morph from time $t = 0$ to time $t = 1$, let $\sigma$ be the time where $b$, $a$, and $e$ become collinear. Observe that $0 < \sigma < 1$ because $b$, $a$ and $e$ are not collinear in $\Gamma_i$ or $\Gamma_{i+1}$. The instant at which they become collinear is indicated by $\overline{b}$, $\overline{e}$ and the dashed line between them in Fig. 13a. We claim that there exists a point $p_{i+1}$ in the interior of $R'$ close enough to $a$ such that $|\overline{p_i p_{i+1}} \cap R| / |\overline{p_i p_{i+1}}| > \sigma$. Namely, if $a$ is not the only point in $R \cap R'$, then $p_{i+1}$ can be chosen as any point of $R \cap R'$ different from $a$. Otherwise, the existence of a suitable point $p_{i+1}$ can be proved by observing that (i) $R'$ is incident to $a$, (ii) the ratio $|\overline{p_i p_{i+1}} \cap R| / |\overline{p_i p_{i+1}}|$ gets arbitrarily close to 1 when $p_{i+1}$ gets arbitrarily close to $a$ (see Fig. 13b).

With this choice of $p_{i+1}$ it remains to show that the morph $\langle \Gamma_i^p, \Gamma_{i+1}^p \rangle$ is planar. Parameterize the morph from time $t = 0$ to time $t = 1$. We are going to prove the following statement: For each line $l$ passing through two consecutive vertices of $C$, vertex $p$ remains on the same side of $l$ at every time instant $0 \leq t \leq 1$ of the morph. Since $p_i$ lies in the kernel of the polygon $C$ in $\Gamma_i^p$, the statement implies that $p$ is in the kernel of the polygon $C$ during the entire morph, which implies that the morph is planar.

We now prove the statement. The 3 cases when $l$ passes through $b$ and $c$, when $l$ passes through $c$ and $d$, and when $l$ passes through $d$ and $e$, easily follow from the definition of $\epsilon$. It remains to consider the cases when $l$ passes through $a$ and $b$ or when $l$ passes through $a$ and $e$. The statement follows easily for these cases if $p_{i+1}$ lies in $R \cap R'$.

Otherwise, note that either $p$ does not intersect the elongation of $(a, e)$ when passing from $p_i$ to $p_{i+1}$ during the morph (as in Fig. 13a) or $p$ does not intersect the elongation

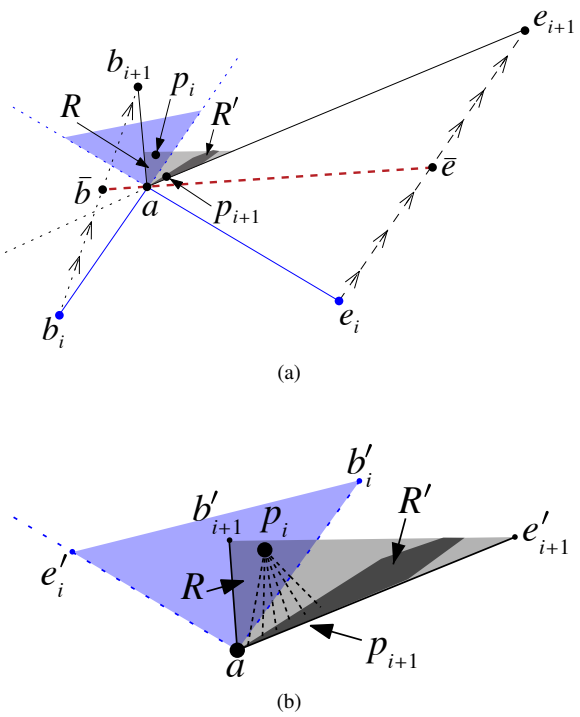



Figure 13: (a) Illustration for the placement of point $p_{i+1}$. The starting $a$-reflex drawing $\Gamma_i$ is blue, as is the corresponding triangle $\tau_i$, while the final $a$-convex drawing $\Gamma_{i+1}$ is black and the corresponding triangle $\tau_{i+1}$ is grey. Point $p_i$ is in region $R = \tau_i \cap \tau_{i+1}$. Point $p_{i+1}$ must be placed in region $R'$. For sake of readability, points $b'$ and $e'$ are not shown. (b) The ratio $|\overline{p_i p_{i+1}} \cap R|/|\overline{p_i p_{i+1}}| > \sigma$ gets arbitrarily close to 1 when $p_{i+1}$ gets arbitrarily close to $a$.

of $(a, b)$ when passing from $p_i$ to $p_{i+1}$. Assume the former, the other case being analogous. Note that this implies the statement when $l$ passes through $a$ and $e$.

Concerning the final case in which $l$ passes through $a$ and $b$, we note that from time $t = 0$ to time $t = \sigma$, $p$ is inside $R$. Since $R$ is delimited by the elongation of $(a, b)$ at time $t = 0$, we have that $p$ is on the same side of $l$ at each time instant $0 \leq t \leq \sigma$. Also, from time $t = \sigma$ to time $t = 1$ vertex $p$ is separated from the elongation of $(a, b)$ by edge $(a, e)$. Since $p_{i+1}$ lies in $R'$, which is delimited by the line through the positions of $a$ and $e$ at time instant $t = 1$, $p$ does not traverse edge $(a, e)$ during the morph. Hence, $p$ remains on the same side of $l$ also at each time instant $\sigma \leq t \leq 1$. This completes the proof of the statement.

To conclude, we have proved by induction on the number of alternations between $a$-convex and $a$-reflex drawings in the sequence $\Gamma_1, \ldots, \Gamma_k$, that we can construct a sequence of drawings $\Gamma_1^p, \ldots, \Gamma_k^p$ that determine a planar morph of the cycle $C$ augmented by vertex $p$ and its edges, as required.

## 6 Conclusion

In this paper we have given an algorithm that takes as input two straight-line planar drawings $\Gamma_1$ and $\Gamma_2$ of the same graph with the same embedding, and finds a morph with a polynomial number of steps from $\Gamma_1$ to $\Gamma_2$ that preserves straight-line planarity. Each step is a linear morph, and the number of steps is $O(n^2)$ and $O(n^4)$ for triangulations and general plane graphs, respectively.

One natural question is to reduce the number of steps. In particular, is there a morph with fewer than $O(n^4)$ steps for general plane graphs? We would need a different approach than compatible triangulation. On the other side, we are not aware of any non-trivial (even linear) lower bound, although we suspect that a constant number of steps do not always suffice.

It would also be interesting to design morphing algorithms for sub-classes of planar graphs, such as *trees*, *outerplane graphs*, and *series-parallel graphs*. We suspect that, for such graph classes, morphs with a linear number of steps always exist. Note that using Lemma 5.1, it is easy to prove that a morph with a linear number of steps exists between any two straight-line planar drawings of a *maximal plane 3-tree*.

Thomassen [33] showed that there exists a morph between any two topologically equivalent planar convex drawings that preserves convexity as well as straight-line planarity. Can this be done with a polynomial number of linear morphs?

It is possible to implement our algorithm in polynomial time under the real RAM model of computation. However, we have not bounded the coordinate values and coefficients in our linear morphs and it seems that they may require a superpolynomial number of bits when converted to integers (though they can be described using 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. We leave as an open problem to find a morph that uses a polynomial number of linear morphing steps and uses only a polynomial (or, even better, a logarithmic) number of bits per coordinate.

**Acknowledgements.** Part of this research was conducted in the framework of ESF project 10-EuroGIGA-OP-003 GraDR “Graph Drawings and Representations" and of ‘EU FP7 STREP Project “Leone: From Global Measurements to Local Management", grant no. 317647’.

Part of this research was done with the support of NSERC, the Natural Sciences and Engineering Research Council of Canada. 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.

## 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 CCCG, 2011.

[2] M. ALEXA, D. COHEN-OR, AND D. LEVIN, *As-rigid-as-possible shape interpolation*, in Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00), 2000, pp. 157–164.

[3] P. ANGELINI, P. F. CORTESE, G. D. BATTISTA, AND M. PATRIGNANI, *Topological morphing of planar graphs*, in Graph Drawing '08, vol. 5417 of Lecture Notes in Computer Science, Springer, 2009, pp. 145–156.

[4] B. ARONOV, R. SEIDEL, AND D. L. SOUVAINE, *On compatible triangulations of simple polygons*, Computational Geometry: Theory and Applications, 3 (1993), pp. 27–35.

[5] G. BAREQUET AND M. SHARIR, *Piecewise-linear interpolation between polygonal slices*, Computer Vision and Image Understanding, 63 (1996), pp. 251–272.

[6] T. BEIER AND S. NEELY, *Feature-based image metamorphosis*, in Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '92), J. J. Thomas, ed., ACM, 1992, pp. 35–42.

[7] T. C. BIEDL, A. LUBIW, AND M. J. SPRIGGS, *Morphing planar graphs while preserving edge directions*, in Graph Drawing '05, vol. 3843 of Lecture Notes in Computer Science, Springer, 2006, pp. 13–24.

[8] S. CAIRNS, *Deformations of plane rectilinear complexes*, The American Mathematical Monthly, 51 (1944), pp. 247–252.

[9] E. CARMEL AND D. COHEN-OR, *Warp-guided object-space morphing*, The Visual Computer, 13 (1997), pp. 465–478.

[10] R. CONNELLY, E. D. DEMAINE, AND G. ROTE, *Straightening polygonal arcs and convexifying polygonal cycles*, Discrete & Computational Geometry, 30 (2003), pp. 205–239.

[11] E. D. DEMAINE AND J. O'ROURKE, *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*, Cambridge University Press, 2007.

[12] C. ERTEN, S. G. KOBOUROV, AND C. PITTA, *Intersection-free morphing of planar graphs*, in 11th Symposium on Graph Drawing, 2003, pp. 320–331.

[13] ———, *Morphing planar graphs*, in 20th ACM Symposium on Computational Geometry, 2004.

[14] H. EVERETT, S. LAZARD, S. M. ROBBINS, H. SCHRÖDER, AND S. WHITESIDES, *Convexifying star-shaped polygons*, in CCCG, 1998.

[15] M. S. FLOATER AND C. GOTSMAN, *How to morph tilings injectively*, Journal of Computational and Applied Mathematics, 101 (1999), pp. 117–129.

[16] C. FRIEDRICH AND P. EADES, *Graph drawing in motion*, Journal of Graph Algorithms and Applications, 6 (2002), p. 2002.

[17] K. FUJIMURA AND M. MAKAROV, *Foldover-free image warping*, Graphical Models and Image Processing, 60 (1998), pp. 100–111.

[18] C. GOTSMAN AND V. SURAZHSKY, *Guaranteed intersection-free polygon morphing*, Computers & Graphics, 25 (2001), pp. 67–75.

[19] L. GUIBAS AND J. HERSHBERGER, *Morphing simple polygons*, in Proceedings of the tenth annual Symposium on Computational Geometry, SCG '94, New York, NY, USA, 1994, ACM, pp. 267–276.

[20] S. G. KOBOUROV AND M. LANDIS, *Morphing planar graphs in spherical space*, Journal of Graph Algorithms and Applications, 12 (2008), pp. 113–127.

[21] A. LUBIW, *Morphing planar graph drawings*, in Canadian Conference on Computational Geometry (CCCG '07), P. Bose, ed., 2007, p. 1.

[22] A. LUBIW AND M. PETRICK, *Morphing planar graph drawings with bent edges*, J. Graph Algorithms and Applications, 15 (2011), pp. 205–207.

[23] A. LUBIW, M. PETRICK, AND M. J. SPRIGGS, *Morphing orthogonal planar graph drawings*, in Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '06), ACM Press, 2006, pp. 222–230.

[24] M. NÖLLENBURG, D. MERRICK, A. WOLFF, AND M. BENKERT, *Morphing polylines: A step towards continuous generalization*, Computers, Environment and Urban Systems, 32 (2008), pp. 248 – 260. Geographical Information Science Research - United Kingdom.

[25] T. W. SEDERBERG, P. GAO, G. WANG, AND H. MU, *2-d shape blending: an intrinsic solution to the vertex path problem*, in Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93), ACM, 1993, pp. 15–18.

[26] T. W. SEDERBERG AND E. GREENWOOD, *A physically based approach to 2-d shape blending*, in Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '92), J. J. Thomas, ed., ACM, 1992, pp. 25–34.

[27] M. SHAPIRA AND A. RAPPAPORT, *Shape blending using the star-skeleton representation*, IEEE Trans. Comp. Graph. Appl., 15 (1995), pp. 44–51.

[28] M. J. SPRIGGS, *Morphing parallel graph drawings*, PhD thesis, School of Computer Science, 2007.

[29] V. SURAZHSKY, *Morphing planar triangulations*, Master's thesis, Technion - Israel Institute of Technology, Israel, 1999.

[30] V. SURAZHSKY AND C. GOTSMAN, *Controllable morphing of compatible planar triangulations*, ACM Transactions on Graphics, 20 (2001), pp. 203–231.

[31] ———, *Intrinsic morphing of compatible triangulations*, International Journal of Shape Modeling, 9 (2003), pp. 191–201.

[32] A. TAL AND G. ELBER, *Image morphing with feature preserving texture*, Comput. Graph. Forum, 18 (1999), pp. 339–348.

[33] C. THOMASSEN, *Deformations of plane graphs*, Journal of Combinatorial Theory, Series B: 34 (1983), pp. 244–257.

[34] W. T. TUTTE, *How to Draw a Graph*, Proceedings of the London Mathematical Society, s3-13 (1963), pp. 743–767.


## Appendix A: Non-linear motion in Cairns's algorithm

The morphing algorithm of Cairns [8] was described in Section 2. Here we give an example to show that Cairns's suggestion of keeping $p$ at the centroid of $\Delta^*(p)$ results in non-linear motion.

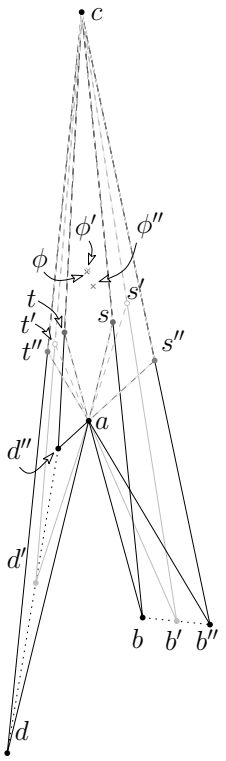



Figure 14: A linear morph of polygon $(a, b, c, d)$, with kernel $a, t, c, s$, shown in three snapshots as $b$ moves to $b'$ and $b''$, and $d$ moves to $d'$ and $d''$. Corner $s$ of $a, t, c, s$ moves along a trajectory that is not a straight line. As a consequence, the centroid $\phi$ of $a, t, c, s$ does not move along a straight line.