# Morphing Orthogonal Planar Graph Drawings*

Anna Lubiw† Mark Petrick† Michael Spriggs†

**Abstract**

We give an algorithm to morph between two planar orthogonal drawings of a graph, preserving planarity and orthogonality. The morph uses a polynomial number of discrete steps. Each step is either a linear morph that moves a set of vertices horizontally or vertically; or a "twist" that introduces new bends in the edges incident with one vertex. Our morph can be implemented so that inter-vertex distances are well-behaved. This is the first algorithm to provide planarity-preserving morphs with well-behaved complexity for a significant class of graph drawings.

## 1 Introduction

A *morph* is a continuous transformation of one object or configuration to another, often with the requirement of preserving some geometric structure. This paper is about morphs of graph drawings. The conventional scenario in graph drawing is to go from an abstract specification of a graph to a geometric drawing, but there are many situations in which the input graph already has a geometric structure. Rather than producing a new drawing "out of the blue," it is desirable to morph from the old drawing to the new one.

An easy way to morph straight-line graph drawings is to move each vertex directly from its initial to its final position. Unfortunately such a *linear morph* may cause edges to intersect even if the initial and final drawings are planar.

When planarity of the drawing must be preserved there are two methods to morph straight-line graph drawings. These methods are described in the background section below. Neither method is ideal: The first, due to Cairns [9] and Thomassen [22] destroys the users mental map by contracting vertices together, and—though it involves discrete steps—comes with no polynomial bound on the number of steps. The second, due to Floater and Gotsman [13], and Gotsman and Surazhsky [17] is not composed of discrete steps; the algorithm computes "snapshots" of the graph drawing at intermediate time points, but no bounds are known on the complexity of the vertex trajectories, nor on the number of snapshots needed to approximate continuous motion.

In this paper we give an algorithm to morph one orthogonal drawing of a graph to another orthogonal drawing of the same graph, maintaining planarity (that edges do not cross) and orthogonality (that each edge is drawn as a path of alternating horizontal and vertical line segments). Our morph is composed of a polynomial number of elementary steps, and we can nicely bound edge shrinkage, inter-vertex distances, and so on. This is the first algorithm to morph a significant class of planar graph drawings via a morph with well-behaved complexity.

Our result builds on a solution to the special case where each edge has the same bends in the same directions in the source and target drawings (Biedl, Lubiw, Spriggs [5]).

## 2 Background

Construed broadly, morphing is an area that includes motion planning, folding problems, linkage reconfiguration problems, rigidity theory, and problems of finding 3-D interpolations between successive 2-D slices. Many of these problems have a mathematical history that predates the coining of the word "morph" in 1975. Morphing is an active area of research in graphics [16]; for a more mathematical treatment of morphing, see [1].

**2.1 Morphing Graph Drawings.** A primary reason to morph graph drawings is for visualization purposes (though different criteria arise when morphing is used to create a 3-D model from 2-D slices [3]). The usual criteria for quality of graph drawings apply to each "frame" of the morph. Even more relevant are the criteria for dynamic graph drawing [7] and interactive graph drawing [3]. Friedrich and Eades [14] formulate two goals for "animation" between two graph drawings: to preserve the mental map (a concept formulated by Misue et al. [20]); and to communicate the structural changes to the graph. They refine these goals to more quantitative criteria such as: minimize edge crossings, use uniform motion as much as possible, etc. They give an animation procedure that partially satisfies their criteria. A subsequent paper by Friedrich and Houle [15]

*supported by NSERC

†School of Computer Science, University of Waterloo

SODA '06, January 22-26, Miami, FL
 ISBN 0-89871-605-5/06/01

enhances this with clustering techniques.

A more mathematical approach to morphing of graph drawings is to study what geometric structure can be preserved during the course of a morph. There have been two significant results on preserving planarity—i.e. ensuring that edges do not cross—which we summarize below. There has also been some work on preserving convexity of faces [22], and directions of edges [4, 6].

Cairns [9] proved in 1944 that there is a planarity-preserving morph from any planar triangulation to any isomorphic one with the same fixed boundary triangle. He shows that there is a vertex $v$ that can be contracted to a neighbour, preserving planarity. Unfortunately, it may be necessary to contract $v$ to a different neighbour in the source and target drawings, so Cairns needs an intermediate step where the neighbourhood of $v$ is morphed to a convex polygon (whereupon $v$ can move from one neighbour to another across the convex face). This means that the algorithm uses two recursive calls on graphs of one fewer vertices, and hence takes exponential time. (Cairns does not analyze his algorithm—this was 1944 after all!) Cairns's morph is not useful for visualization purposes; over the course of the morph, the graph contracts to a triangle and then re-emerges.

Thomassen [22] extended Cairns' result to prove that there is a planarity-preserving morph between any two straight-line drawings of a planar graph with the same embedding (i.e. the same faces and the same outer face). He reduces to Cairns' result using the technique of "compatible triangulation" (discovered independently by Aronov et al. [2]) to augment both drawings to isomorphic triangulations with the same fixed boundary triangle. This morph inherits the same flaws: it takes exponential time, and contracts the graph down to a triangle.

An alternative approach to planarity-preserving morphs of triangulations was developed by Floater and Gotsman [13]. They use Tutte's planar graph drawing technique of barycentric coordinates, and prove that a linear morph of the matrix of barycentric coordinates yields a planarity-preserving morph of the triangulation. Explicit vertex trajectories are not computed; rather the algorithm efficiently computes a snapshot of the graph at any specified intermediate time point. The choice of time points to approximate continuous motion is not explored. Gotsman and Surazhsky [17, 21] combine this result with the compatible triangulation result of Aronov et al. [2] to give an alternative to Thomassen's planarity-preserving morph of straight-line planar graph drawings. This morph is visually appealing, but comes without time or quality guarantees. Besides the issue of the number of time points required, there is also the issue—as in Tutte's original drawing method—of how close vertices may get to one another. Erten et al. [12, 11] have implemented the Floater Gotsman approach, with a preliminary rigid motion stage.

**2.2 Orthogonal Graph Drawing.** There is an extensive body of work on orthogonal graph drawing. For a good survey see Eiglsperger, Fekete, and Klau [10]; we do not attempt a summary here. Although morphing of orthogonal graph drawings has not been explored, many orthogonal graph drawing methods work by adjusting an existing drawing, for example: compaction techniques [18], bend-reduction techniques (see [10]), and interactive drawing methods [8].

## 3 Preliminaries

An *orthogonal drawing* of a graph $G = (V, E)$ assigns to each vertex $v \in V$ a distinct point $p(v)$ in the plane, and to each edge $e = (u, v)$ a path from $p(u)$ to $p(v)$ of $k_e$ line segments, alternately horizontal and vertical. We require that no two line segments in the drawing overlap in an interval of positive length. As a consequence, a graph with a vertex of degree more than 4 has no orthogonal drawing. An orthogonal drawing is *planar* if the paths representing edges are disjoint except at common endpoints.

The $k_e - 1$ points between consecutive segments along the path representing edge $e = (u, v)$ are called *bends*. Traversing the path representing $e$ from $u$ to $v$, we identify each bend as a left turn or a right turn, and we identify the *direction* of the $i^{\text{th}}$ segment, $d^i(e)$, as either $N, S, E$ or $W$. Note that these definitions depend on whether we traverse the path from $u$ to $v$ or from $v$ to $u$. When we say "an edge $e = (u, v)$" we mean that we will traverse its path from $u$ to $v$. When we say "all edges" we include $(u, v)$ and $(v, u)$. For edge $e = (u, v)$, we define the *direction sequence* of the edge, $d(e)$ to be the sequence $d^1(e), d^2(e), \ldots, d^{k_e}(e)$, and we call $d^1(e)$ the *initial direction* of the edge. We will distinguish different drawings via subscripts; thus $d^1_P(e)$ is the initial direction of edge $e$ in drawing $P$.

The *size*, $n_P$ of an orthogonal graph drawing $P$ is the sum of the number of vertices, number of edges, and number of bends. We will analyze the complexity of a morph between two orthogonal graph drawings $P$ and $Q$ in terms of $n$, the maximum of $n_P$ and $n_Q$.

A *morph* between two drawings $P$ and $Q$ of the same graph $G$ 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 $P$ and the drawing at time $t = 1$ is $Q$. A morph preserves planarity [orthogonality] if all intermediate drawings are planar [orthogonal]. It is important to note that in order to have a planarity preserving morph between two

drawings $P$ and $Q$ of graph $G$, the two drawings must represent the same planar embedding of $G$—i.e. have the same cyclic ordering of edges around vertices, and the same outer face.

We say that two planar orthogonal drawings $P$ and $Q$ of a graph $G$ are *parallel* if every edge has the same direction sequence in $P$ and in $Q$. In particular $P$ and $Q$ must have the same set of bends.

In this paper we morph between orthogonal drawings that are not parallel, and so we must be able to create and eliminate bends. We will therefore allow intermediate drawings in which a path representing an edge has line segments of length 0, where a bend becomes coincident with another bend or vertex. We do not consider this a violation of planarity.

Given parallel orthogonal drawings $P$ and $Q$ of the same graph $G$ the *linear morph* moves each vertex and bend in a straight line at uniform speed from its position in $P$ to its position in $Q$; the path respresenting each edge is determined in the obvious way. It is easy to prove that the linear morph preserves orthogonality but not necessarily planarity.

## 4 The Morph

This section contains our main result, an algorithm to morph from one orthogonal drawing $P$ of a graph $G = (V, E)$ to another orthogonal drawing $Q$ of the same graph while preserving planarity and orthogonality. We assume that the two drawings represent the same embedding of $G$, otherwise there is no such morph. In the final stage of our morph we make use of an algorithm by Biedl, Lubiw, Spriggs [5] for the special case when $P$ and $Q$ are parallel. In the first three stages of our morph we make successive progress toward making $P$ and $Q$ parallel. Each stage must maintain the properties achieved by the previous stages.

1. Morph $P$ so that for any edge $e = (u, v)$, the initial direction of $e$ at $u$ is the same in $P$ as it is in $Q$—i.e. $d^1_P(e) = d^1_Q(e)$.

2. Morph $P$ so that no edge $e$ has a *zig-zag*—a left turn followed by a right turn, or vice versa. Similarly, morph $Q$ to eliminate zig-zags. A morph that eliminates a zig-zag was given by Biedl, Lubiw, Spriggs [5].

3. At this point each edge that isn't straight must *spiral* with only left turns or only right turns. Morph $P$ so that each edge spirals the same number of times in the same direction in $P$ as in $Q$.

4. At this point $P$ and $Q$ are parallel; apply the algorithm of Biedl, Lubiw, Spriggs [5].

Note that although we describe the morph as operating sometimes on $P$ and sometimes on $Q$, the final morph simply goes from $P$ to $Q$. More specifically, if morph $\rho$ transforms $P$ to $R$ and morph $\tau$ transforms $Q$ to $R$, then the morph composed of $\rho$ followed by the reverse of $\tau$ transforms $P$ to $Q$.

In the remainder of this section we treat stages 1–3 in turn.

### 4.1 Altering Initial Edge Directions.

The first stage of the morph alters the directions in which edges are incident to their endpoints in $P$ so that they match $Q$.

Consider first the case when only one edge is incident to $u$. We can alter $d^1_P(e)$ by adding 3 bends to the segment as shown in Figure 1. This continuous motion changes $d^1_P(e)$ by $\pi/2$ radians and preserves orthogonality. If we keep the bends close enough to $u$, planarity is also preserved. We call this motion a *twist*. The twist shown in Figure 1 is a *clockwise* twist.

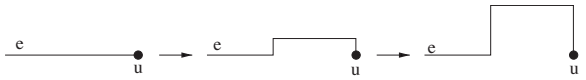


Figure 1: A morph that alters the initial direction of edge $e$ at vertex $u$ from $W$ to $N$ by adding 3 bends.

More generally, when $u$ has more than one incident edge, we can simultaneously apply a twist to any set of edges incident with $u$ so long as the new edge directions do not conflict with any unaltered edge directions. In particular, we can twist all the edges incident with $u$, as shown for the case of a counter-clockwise twist in Figure 2. One twist changes initial edge directions by $\pi/2$ radians either clockwise or counterclockwise. A sequence of two twists can be used to change initial edge directions by $\pi$ radians; see the middle pane of Figure 2. Since $P$ and $Q$ represent the same planar embedding, the cyclic order of edges around vertex $u$ is the same in $P$ and $Q$, and there exists a sequence of at most 4 clockwise (or counterclockwise) twists that will orient the edges around vertex $u$ in $P$ to match $Q$. This proves the following lemma:

LEMMA 4.1. *Let $P$ and $Q$ be planar orthogonal drawings of the same graph embedded in the same way. Drawing $P$ can be morphed by applying $O(n_P)$ twists so that initial edge directions match those in $Q$. The morph preserves planarity and orthogonality. The resulting drawing has size $O(n_P)$.*

In the third stage of the algorithm we will make use of a *full twist* that consists of four twists in sequence;

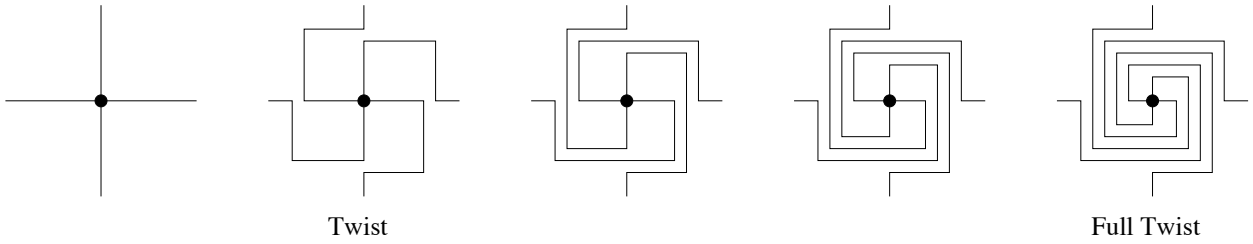


Figure 2: Applying a sequence of counter-clockwise twists to all edges incident to a vertex.

see the last pane of Figure 2. A full twist causes edges to spiral while leaving initial edge directions unchanged.

**4.2 Eliminating Zig-Zags.** In stage 2 of the morph we eliminate zig-zags in $P$ and in $Q$ using the technique from [5]. Each zig-zag is eliminated using a linear morph that we call a *slide*. A slide introduces no new bends, leaves initial edge directions fixed, and preserves planarity and orthogonality. We thus obtain the following lemma.

LEMMA 4.1. *A planar orthogonal graph drawing $P$ can be morphed to eliminate zig-zags using $O(n_P)$ slides. The morph preserves planarity and orthogonality. The resulting drawing has size $O(n_P)$.*

We mention that the term "zig-zag" was used by Liu, Morgana and Simeone [19]; they eliminate zig-zags in orthogonal graph drawings, although only in a special situation, and not via a morph.

Although the zig-zag elimination technique comes from [5], we describe it here for the sake of completeness. Let $e \in E$ be an edge that is drawn in $P$ containing a zig-zag, and let $\alpha\beta\gamma\delta$ be the sequence of vertices/bends involved in this zig-zag. We use the same notation $\alpha$, $\beta$, $\gamma$, $\delta$ for a vertex/bend and for the point in the plane at which it is drawn. We will describe the zig-zag elimination for the case where $\alpha$, $\beta$, $\gamma$, $\delta$ are in the configuration shown in Figure 3(a). The other cases are rotated/flipped versions of this. Note that $\beta$ and $\gamma$ must be bends; $\alpha$ and $\delta$ may be bends or vertices. Consider the subset $\mathcal{V}$ of vertices/bends in $P$ that lie either:

1. *strictly* above the horizontal ray originating at $\beta$ and going leftward; or
2. on or above the horizontal ray originating at $\gamma$ and going rightward (including $\gamma$ but not $\beta$).

Morph $P$ by moving all vertices in $\mathcal{V}$ upward a distance equal to the initial distance between $\beta$ and $\gamma$, with the movement occuring at a uniform rate. Keep all other vertices/bends stationary. Call the resulting drawing $P'$; see Figure 3(b). We call this linear morph from $P$ to $P'$ a *slide* on zig-zag $\alpha\beta\gamma\delta$.

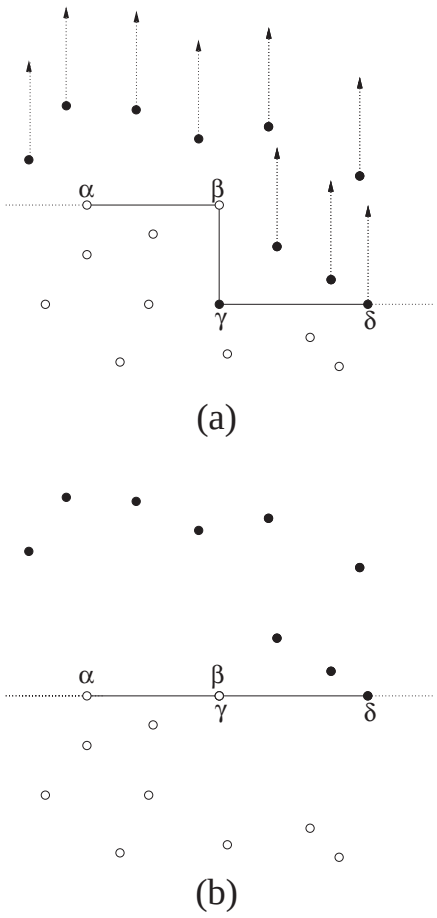


Figure 3: Before (a) and after (b) the slide for zig-zag elimination.

In $P'$, $\beta$ coincides with $\gamma$. Further, $\alpha$ and $\delta$ share the same vertical coodinate. Hence, in $P'$ the zig-zag $\alpha\beta\delta\gamma$ can be replaced by a horizontal segment between $\alpha$ and $\gamma$, thereby removing one right turn and one left turn from the drawing of the edge.

Figure 3 shows a *vertical slide*, but we would use a *horizontal slide* if $\beta, \gamma$ is a horizontal segment.

LEMMA 4.2. *The slide removes a single zig-zag from $P$ while keeping intermediate drawings both planar and orthogonal.*

*Proof.* We have already described how the zig-zag is removed. It remains to show that all intermediate drawings of the slide are planar and orthogonal.

Since movement is along the vertical axis, vertical segments remain parallel with the vertical axis throughout the morph. Horizontal segments with both endpoints in $\mathcal{V}$ or both endpoints outside $\mathcal{V}$ remain horizontal throughout the morph. There are no horizontal segments with one endpoint in $\mathcal{V}$ and one endpoint outside $\mathcal{V}$, because such a segment would cross segment $(\beta, \gamma)$.

It remains to show that planarity is preserved; observe that if a segment moves upward during the slide, then any segment above it that it might potentially intersect moves upward by the same amount and at the same rate.

**4.3 Adjusting Spirality.** After the first two stages of the algorithm the initial direction of any edge is the same in $P$ and $Q$ and no edge in either drawing contains both left and right turns. We say that an edge with only left turns [or right turns] *spirals*. In stage three, we will show how to morph $P$ so that each edge spirals in the same direction and with the same number of bends in $P$ as in $Q$.

Let $l(e)$ and $r(e)$ denote the number of left and right turns [respectively] along edge $e = (u, v)$ traversed from $u$ to $v$, and let $s(e) = l(e) - r(e)$. When $e$ has only left [or right] turns, $s(e)$ is the (signed) *spirality* of edge $e$. As usual, we distinguish these measures in the two drawings using subscripts $P$ and $Q$. Let $S(e) = s_P(e) - s_Q(e)$ be a comparison of the spirality of $e$ in $P$ and $Q$. Observe that for any edge $e$, $S(e) \equiv 0 \pmod 4$ because the initial directions of edges are the same in $P$ and $Q$. We want to morph so that $S(e)$ becomes 0 for all edges $e$.

LEMMA 4.2. *Let $P$ and $Q$ be planar orthogonal drawings of the same graph embedded in the same way. Suppose that the initial direction of every edge is the same in $P$ and in $Q$. Suppose that every edge spirals in both drawings. We can morph $P$ so that $S(e)$ becomes 0 for all edges $e$ using $O(n^2)$ twist operations and $O(n^2)$ slide operations. The morph preserves planarity and orthogonality.*

Our basic step is to use a full twist on some set of vertices to improve the spirality of one edge by 4. Since the graph has $O(n)$ bends, the spirality measures are $O(n)$; thus the number of basic steps required is $O(n)$, and the lemma above is proved by:

LEMMA 4.3. *Under the same hypotheses as the Lemma above, and supposing that $S$ is not identically 0, there is a morph of $P$ using $O(n)$ twists and $O(n)$ slides that reduces $|S(e)|$ for some edge $e$ and does not increase $|S(f)|$ for any other edge $f$. The morph preserves planarity and orthogonality and the property that every edge spirals. Furthermore, the size of the resulting drawing is $O(n)$, where $n$ is the size of the* original *input drawings.*

*Proof.* Our main tool is the full twist. Observe that for edge $e = (u, v)$, performing a clockwise full twist at vertex $v$ and then eliminating the resulting zig-zags decreases $s(e)$ by 4. We begin by assigning directions to edges to record which endpoint of the edge should be twisted clockwise to improve the spirality. If edge $e = (u, v)$ has $S(e) > 0$ then direct $e$ from $u$ to $v$. If edge $e = (u, v)$ has $S(e) < 0$ then direct $e$ from $v$ to $u$. Edges with $S(e) = 0$ remain undirected. Then, for an edge $e$ directed from $x$ to $y$, a clockwise full twist at $y$ in drawing $P$ reduces $|S(e)|$ by 4. There is one difficulty: the twist at $y$ will make spirality worse for any directed edge leaving $y$, and for any undirected edge incident with $y$—we must also twist at the other ends of such edges. To prove that we can make progress we need an argument about directed cycles in the graph. A cycle is *directed* if it does not go backwards along a directed edge—it may use directed and/or undirected edges.

PROPOSITION 4.1. *With the above assignment of directions to edges there is no directed cycle that contains at least one directed edge.*

*Proof.* By contradiction. Consider such a cycle $C$. Because both $P$ and $Q$ are planar drawings of the graph, $C$ is a simple cycle in both drawings and has the same inside/outside. Define $s(C) \equiv \sum_{e \in C} s(e)$. We will count $s_P(C)$ and $s_Q(C)$ as we go around $C$. As proved by Vijayan and Wigderson [23], any planar orthogonal cycle has 4 more right bends than left if traversed clockwise (and 4 more left bends than right if traversed counterclockwise). Hence $s_P(C) = s_Q(C)$. Since the initial direction of each edge is the same in $P$ and $Q$, bends that occur at a vertex of the cycle are the same in $P$ and $Q$. Also, bends that occur on an undirected edge $e$ are the same in $P$ and $Q$ since $S(e) = 0$. The only other source of bends is along directed edges. Along a directed edge $e$, $S(e) > 0$, so $s_P(e) > s_Q(e)$. Because $C$ respects edge directions and contains at least one directed edge, thus $s_P(C) > s_Q(C)$. Contradiction.

We continue the proof of the Lemma, making use of the proposition to identify a set of vertices to twist. Ignoring the drawings for the moment, consider the graph with the edges directed as described above. Define $G_R$ to be the graph formed by contracting together any vertices connected by an undirected edge. A vertex in $G_R$ corresponds to a connected component formed by the undirected edges in the original graph. The above claim proves that $G_R$ is acyclic (see Figure 4).

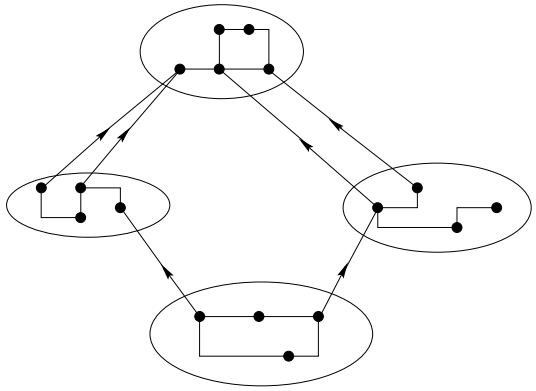

Figure 4: The graph $G_R$ is a directed acyclic graph.

Take a sink of $G_R$ (i.e. a vertex with no out-going edges) and let $U$ be the vertices of $G$ in this sink. In drawing $P$, apply clockwise full twists to all vertices in $U$ and then eliminate the resulting zig-zags. See Figure 5 for an example. We claim that this morph satisfies the requirements of the Lemma.

The morph strictly improves spirality (i.e. reduces $|S(e)|$) on any edge $e$ directed in to $U$, and there is at least one such edge. No edge is directed out of $U$. Undirected edges with one endpoint in $U$ must have both endpoints in $U$, and twisting at both ends of an edge $f$ leaves $|S(f)|$ unchanged.

It remains to consider the size of the resulting drawing. Twisting the vertices of $U$ adds $O(n)$ bends to the graph, which is potentially dangerous. However, zig-zag elimination removes bends. The crucial observation is that since $|S(f)|$ never increases for any edge $f$, thus after we have eliminated zig-zags to make every edge spiral, the number of bends along edge $f$ is always between the original value of $|s_P(f)|$ and the value of $|s_Q(f)|$.

**4.4 Analysis of the Morphing Algorithm.** In this section we analyze the number of elementary operations—twists and slides—used by our morph. Stage 4 of the morph uses one other elementary operation, a linear morph. This linear morph is performed on a "rectangular" graph, and can therefore be separated into its horizontal and vertical components; these are slightly different from slides in that vertices may move by different amounts.

There are then two kinds of elementary operations required by our morphs: twists and linear morphs that move a set of vertices horizontally or vertically.

LEMMA 4.4. *Given two planar orthogonal drawings $P$ and $Q$ of the same graph embedded in the same way, there is a morph from $P$ to $Q$ that preserves planarity and orthogonality and is composed of $O(n^2)$ twists, $O(n^2)$ slides, and one linear morph, where $n$ is the maximum of the size of $P$ and the size of $Q$.*

*Proof.* By Lemmas 4.1, 4.1, and 4.2, stages 1–3 of the morph take a total of $O(n^2)$ twists and $O(n^2)$ slides, and the resulting drawings have size $O(n)$. Stage 4 takes $O(n)$ slides and one linear morph [5].

Observe that a single slide moves $O(n)$ vertices so describing vertex trajectories explicitly takes $O(n)$ space. This raises the space required to explicitly describe the morph to $O(n^3)$.

The morph, as described in this section, does not behave well in terms of edge-lengths and inter-vertex distances. In particular, stage 4, which involves only slides, and thus only increases edge lengths, may increase them exponentially [5]. Additionally, twists introduce new grid lines, and doing this carelessly and in combination with slides may cause vertices to move very close together. We remedy this in the following section.

## 5 Quality of the Morph

**5.1 Feature Size.** As mentioned in the previous section 4.4 the morph, as specified so far, is not well-behaved with respect to edge lengths or *feature size*—the minimum distance between two vertices, or between a vertex and a non-incident edge. In this section we describe a way of implementing the morph to remedy this. The main idea is a general one that was used by Biedl, Lubiw, and Spriggs [5]: Extract from the morph the time points where the vertex orderings change. Draw the graph at each such time point on a small grid. Use a linear morph to go between drawings for consecutive time points. Using this technique we obtain the following result:

THEOREM 5.1. *Let $P, Q$ be a pair of planar orthogonal drawings of a graph $G$ such that all vertices/bends of $P$ and $Q$ lie on a unit grid within a $O(n) \times O(n)$ bounding box, where the size of both $P$ and $Q$ is at most $n$. There exists a morph from $P$ to $Q$ that is composed*

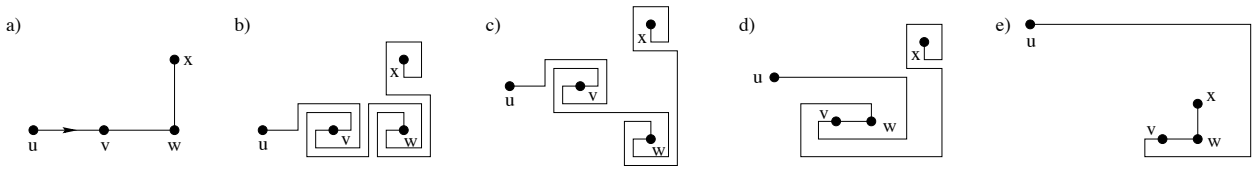


Figure 5: Adjusting spirality on edge $(u, v)$: (a) $U = \{v, w, x\}$; (b) applying full twists to vertices of $U$; (c)–(d) eliminating zig-zags; (e) the result.

*of a sequence of $O(n^4)$ linear morphs, such that all intermediate drawings are planar and orthogonal and lie in an $O(n) \times O(n)$ bounding box and have a minimum feature size of $1/\sqrt{2}$.*

Due to space limitations we do not give a detailed proof. As described above, we identify stretches of time during the morph when the $x$ and $y$ orderings of vertex coordinates do not change (i.e. the relations $<$ and $=$ do not change). We pick one time point from each such stretch of time. Thus from one time point to the next the two drawings are parallel and the orderings of vertex coordinates change only in that for some pairs $a, b$ of $x$ [or $y$] vertex coordinates it may happen that $a = b$ becomes $a < b$ or vice versa. We say that two consecutive time points have *weakly equivalent* vertex orderings. Drawing the graph for each time point on a small grid is easy (it is important that the intermediate graphs have size $O(n)$). The main thing we must justify is that a linear morph between the drawings for two consecutive time points preserves planarity and has the required minimum feature size—see Lemma 5.1 below. The other thing worth addressing is the number of time points. Recall that our morph uses $O(n^2)$ twists and $O(n^2)$ slides. A twist introduces new bends, but produces only $O(1)$ time points. During a slide, however, there may be $O(n)$ vertices sliding past $O(n)$ vertices, resulting in a total of $O(n^2)$ time points.

The following lemma was claimed by Biedl, Lubiw and Spriggs [5]. For the sake of completeness we give a proof.

LEMMA 5.1. *Let $P$ and $Q$ be parallel planar orthogonal drawings of graph $G$ such that all vertices/bends of $P$ and $Q$ lie on a unit grid and $P$ and $Q$ have weakly equivalent vertex orderings. A linear morph between $P$ and $Q$ preserves orthogonality and planarity, and maintains a minimum feature size of at least $1/\sqrt{2}$.*

*Proof.* The minimum feature size is achieved by either the distance between two vertices or the distance between a vertex and a non-incident edge. We give the argument for the distance between two vertices; the other case is similar.

Consider two vertices $u$ and $v$. Suppose they are drawn at points $p(u), p(v)$ in $P$ and at points $q(u), q(v)$ in $Q$, where $p(u) = (p_x(u), p_y(u))$ and so on. Without loss of generality assume drawing $P$ puts $u$ in the upper right quadrant compared to $v$, i.e. that $p_x(u) \geq p_x(v)$ and $p_y(u) \geq p_y(v)$. Since the vertex orderings are weakly equivalent in $P$ and $Q$ this implies that $q_x(u) \geq q_x(v)$ and $q_y(u) \geq q_y(v)$.

Let $r_t(u) = tq(u) + (1-t)p(u)$ be the position of vertex $u$ at any time $t \in [0, 1]$ during the linear morph from $P$ to $Q$, and similarly for $r_t(v)$.

We want a lower bound on the $L_2$-distance between $r_t(u)$ and $r_t(v)$ at any time $t \in [0, 1]$. Recall that the $L_1$-distance $||.,.||_1$ between two points $(x_1, y_1)$ and $(x_2, y_2)$ is defined as $|x_1 - x_2| + |y_1 - y_2|$. We make use of the fact that $L_2$- and $L_1$-distances are related by the inequality $||.,.||_2 \geq \frac{1}{\sqrt{2}}||.,.||_1$. Thus it suffices to have a lower bound on the $L_1$-distance between $r_t(u)$ and $r_t(v)$.

For any $0 \leq t \leq 1$,

$$
\begin{aligned}
&||r_t(u), r_t(v)||_1 \\
&= |tq_x(u) + (1-t)p_x(u) - tq_x(v) - (1-t)p_x(v)| \\
&\quad + |tq_y(u) + (1-t)p_y(u) - tq_y(v) - (1-t)p_y(v)| \\
&= t(q_x(u) - q_x(v)) + (1-t)(p_x(u) - p_x(v)) \\
&\quad + t(q_y(u) - q_y(v)) + (1-t)(p_y(u) - p_y(v)) \\
&\quad \text{by order assumption} \\
&= t||q(u), q(v)||_1 + (1-t)||p(u), p(v)||_1 \\
&\geq 1 + (1-t) = 1
\end{aligned}
$$

where the last inequality is based on the assumption that $u$ and $v$ are drawn at distinct points of the unit grid in $P$ and in $Q$.

**5.2 Appearance.** Our focus has been on complexity rather than aesthetics. It remains to be seen whether our morph can be implemented in a visually satisfying way.

The example in Figure 5 is not promising in this regard. We offer one other example. Gotsman and Surazhsky [17] demonstrated their morph transforming the letter S to the letter U. Their illustration, taken

from [17], is shown in Figure 6. The result of our morph on an orthogonal version of the input is shown in Figure 7. This example was done by hand; we did some scaling, and we eliminated spirals using successive half-twists rather than a full twist (see lines (c) and (d) of the figure).

One thing to observe is that the twists, which seem to clutter up the drawings with many strange little bends, actually guide the large-scale movement of the vertices. It may be possible to implement our morph in a way that visually emphasizes the large-scale movements.

A general issue for morphing is whether it is desirable to morph via some "canonical" form, or whether it is preferable to minimize changes between the source and target. We use the second approach in stage 3 when we adjust spirality. However, the zig-zag elimination of stage 1 follows the first approach: it might happen that some edge has the same sequence of turns in the source and target and our algorithm undoes them all. An alternative would be to compare the turn sequence of each edge in the source and the target, and try to preserve common subsequences while morphing. The most expedient way to do this is to introduce new vertices at bends along the edge.

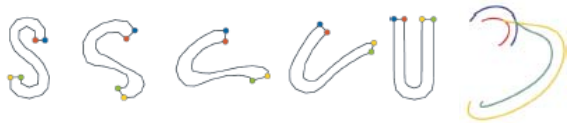

Figure 6: Morphing the letter S to the letter U, Gotsman and Surazhsky [17].

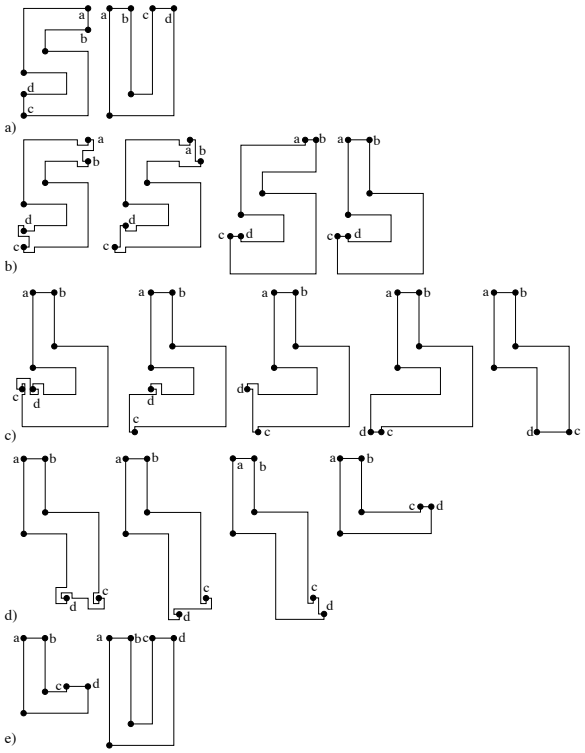

Figure 7: Morphing the letter S to the letter U: (a) Source and target drawings; (b) Adjusting initial directions in source, then straightening zigzags; (c) and (d) Twisting vertices c and d, performed in two phases of twisting 180° and straightening zigzags; (e) parallel morph to target drawing.

## 6 Conclusion

We have presented an efficient way to morph between two planar orthogonal drawings of a graph while maintaining planarity and orthogonality. The morph can be implemented to be well-behaved in the sense that intermediate drawings have small area without vertices/edges getting too close together. It remains to be seen whether our morphs are visually satisfying.

This is the first efficient, well-behaved, planarity preserving morph for a signifiant class of graph drawings. It is an open problem to devise such morphs for all planar graph drawings.

## References


[1] H. Alt and L. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 121 – 153. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 1999.

[2] B. Aronov, R. Seidel, and D. Souvaine. On compatible triangulations of simple polygons. *Computational Geometry: Theory and Applications*, 3:27–35, 1992.

[3] G. Barequet and M. Sharir. Piecewise-linear interpolation between polygonal slices. *Comput. Vis. Image Underst.*, 63(2):251–272, 1996.

[4] T. Biedl, A. Lubiw, and M. Spriggs. Parallel morphing of trees and cycles. In *15th Canadian Conference on Computational Geometry (CCCG)*, pages 29–32, 2003.

[5] T. Biedl, A. Lubiw, and M. Spriggs. Morphing planar graphs while preserving edge directions, 2005. submitted to Graph Drawing 2005.

[6] T. Biedl, A. Lubiw, and M. J. Spriggs. Angles and lengths in reconfigurations of polygons and polyhedra. *Proc. Mathematical Foundations of Computer Science (MFCS'04)*, pages 748–759, 2004.

[7] J. Branke. Dynamic graph drawing. In D. Kaufmann and D. Wagner, editors, *Drawing Graphs – Methods and Models, Lecture Notes in Computer Sci ecne 2025*,

pages 228–246. Springer, 2001.

[8] S. Bridgeman and R. Tamassia. Difference metrics for interactive orthogonal graph drawing algorithms. *Journal of Graph Algorithms and Applications*, 4 (3):47–74, 2000.

[9] S.S. Cairns. Deformations of plane rectilinear complexes. *American Math. Monthly*, 51:247–252, 1944.

[10] M. Eiglsperger, S.P. Fekete, and G.W. Klau. Orthogonal graph drawing. In D. Kaufmann and D. Wagner, editors, *Drawing Graphs – Methods and Models, Lecture Notes in Computer Sciecne 2025*, pages 121–171. Springer, 2001.

[11] C. Erten, S. Kobourov, and C. Pitta. Intersection-free morphing of planar graphs. In G. Liotta, editor, *Graph Drawing 2003, Lecture Notes in Computer Science 2912*, pages 320–331. Springer, 2004.

[12] C. Erten, S. Kobourov, and C. Pitta. Morphing planar graphs. In *20th Annual ACM Symposium on Computational Geometry*, pages 451–452, 2004.

[13] M. Floater and C. Gotsman. How to morph tilings injectively. *Journal of Computational and Applied Mathematics*, 101:117–129, 1999.

[14] C. Friedrich and P. Eades. Graph drawing in motion. *Journal of Graph Algorithms and Applications*, 6 (3):353–370, 2002.

[15] C. Friedrich and M.E. Houle. Graph drawing in motion II. In P. Mutzel, M. Jünger, and S. Leipert, editors, *Graph Drawing 2001, Lecture Notes in Computer Science 2265*, pages 220–231. Springer, 2002.

[16] J. Gomes, L. Darsa, B. Costa, and L. Velho. *Warping and Morphing of Graphical Objects*. Morgan Kaufmann, 1999.

[17] C. Gotsman and V. Surazhsky. Guaranteed intersection-free polygon morphing. *Computers and Graphics*, 25:67–75, 2001.

[18] A.S. LaPaugh. VLSI Layout Algorithms. In M. Atallah, editor, *Algorithms and Theory of Computation Handbook*. CRC Press, 1998.

[19] Y. Liu, A. Morgana, and B. Simeone. A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid. *Discrete Applied Mathematics*, 81(1-3):69–91, 1998.

[20] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Lang. Comput.*, 6 (2):183–210, 1995.

[21] V. Surazhsky and C. Gotsman. Controllable morphing of compatible planar triangulations. *ACM Transactions on Graphics*, 20(4):203–231, 2001.

[22] C. Thomassen. Deformations of plane graphs. *Journal of Combinatorial Theory*, Series B: 34:244–257, 1983.

[23] G. Vijayan and A. Wigderson. Rectilinear graphs and their embeddings. *SIAM J. on Computing*, 14(2):355–372, 1985.