Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 17, no. 5, pp. 575–595 (2013)
DOI: 10.7155/jgaa.00305

Smooth Orthogonal Layouts

M. A. Bekos[1] *M. Kaufmann*[1] *S. G. Kobourov*[2] *A. Symvonis*[3]

[1]Institute for Informatics, University of Tübingen, Germany
[2]Department of Computer Science, University of Arizona, USA
[3]School of Applied Mathematics and Physical Sciences, NTUA, Greece

Abstract

We study the problem of creating *smooth orthogonal layouts* for planar graphs. While in traditional orthogonal layouts every edge is made of a sequence of axis-aligned line segments, in smooth orthogonal layouts every edge is made of axis-aligned segments and circular arcs with common tangents. Our goal is to create such layouts with low edge complexity, measured by the number of line and circular arc segments. We show that every 4-planar graph has a smooth orthogonal layout with edge complexity 3. If the input graph has a complexity-2 traditional orthogonal layout, we can transform it into a smooth complexity-2 layout. Using the Kandinsky model for removing the degree restriction, we show that any planar graph has a smooth complexity-2 layout.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
November 2012	May 2013	October 2013	October 2013	October 2013

Published: October 2013

Article type: Regular paper — Communicated by: G. Di Battista

This article is based on the preliminary version [M. A. Bekos, M. Kaufmann, S. G. Kobourov, A. Symvonis, Smooth Orthogonal Layouts, in: Proc. 20th Int. Symposium on Graph Drawing (GD2012), in: Lecture Notes in Computer Science, Volume 7704, p.p. 150-161, 2013]. The work of M.A. Bekos is implemented within the framework of the Action "Supporting Postdoctoral Researchers" of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State. Michael Kaufmann has been supported by Deutsche Forschungsgemeinschaft SPP 1335 'Scalable Visual Analytics'. The work of S. G. Kobourov is supported in part by NSF grant CCF-1115971 and a grant from the Humboldt Foundation.

E-mail addresses: bekos@informatik.uni-tuebingen.de (M. A. Bekos) mk@informatik.uni-tuebingen.de (M. Kaufmann) kobourov@cs.arizona.edu (S. G. Kobourov) symvonis@math.ntua.gr (A. Symvonis)

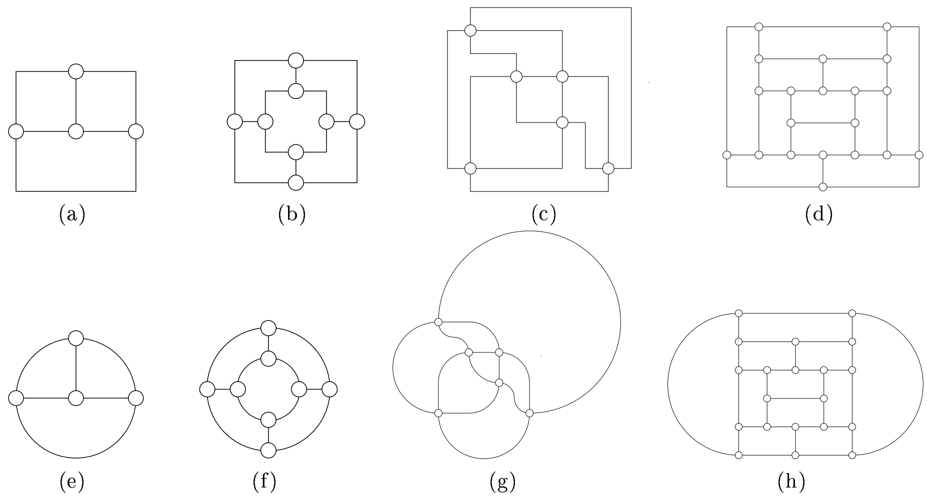


Figure 1: All Platonic solids with degree 3 or 4 drawn in traditional orthogonal style with the minimum number of bends per edge and redrawn in the smooth orthogonal style with better edge complexity.

1 Introduction

Orthogonal graph drawing has a long tradition, dating back to VLSI layouts and floor-planning applications [3, 30, 34, 35, 36]. If the input graph is planar, then it usually required that the output drawing is planar, as well. Even in cases where the input graph is not planar, there exist common techniques, e.g., the *planarization phase* of the *topology-shape-metrics* approach [34], in which a "planar embedding" is computed for a given non-planar graph by replacing the edge crossings by dummy vertices. Hence, *4-planar graphs*, i.e., planar graphs with maximum degree at most four, play an important role in the field of orthogonal graph drawing and arise in a natural way due to the port restrictions. In particular, the goal is to produce a drawing in which each vertex is a point on the integer grid and each edge is represented by a sequence of horizontal and vertical line segments, while optimizing various features of the layout. Typical desirable features include minimizing the used area [35] and minimizing the total number of bends [21, 34], or, the maximum number of bends per edge [2]. Finding an embedding with the minimum number of bends is an NP-hard problem [22]; moreover, minimizing the total number of bends might lead to some edges with many bends. The readability of poly-line drawings decreases as the number of bends increases and the bend angles decrease. One explanation is that every bend interrupts the eye movement and requires a change of direction, with the effect depending on the magnitude of the bend angle.

We hope that, in most cases, by replacing poly-line edges with smooth curves (e.g., composed of two or more circular arcs with common tangents) results in layouts with improved readability and/or more aesthetic appeal; see Figure 1.

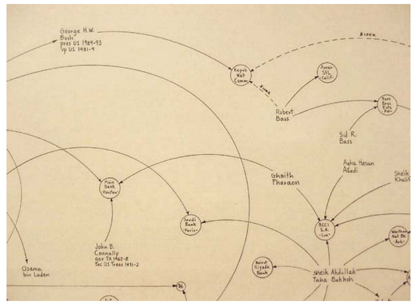


Figure 2: Part of a Mark Lombardi drawing.

Formally, a *smooth orthogonal layout* of a 4-planar graph G is one where (i) each vertex of G is drawn as a point on the plane; (ii) each edge of G is drawn as a sequence of axis-aligned line-segments and circular arc-segments, such that consecutive segments in the sequence have a common point of intersection and a common tangent at that point that is either horizontal or vertical; (iii) there are no edge-crossings; and (iv) there cannot be two segments incident to the same vertex using the same port. Notice that, using the same port necessarily causes overlaps only in the case of straight-line segments. Hence, in principle one can accept multiple arc-segments incident to a vertex from the same direction if they have different radii. We say that a smooth orthogonal layout is of *edge complexity* $k \geq 1$, if it contains an edge of complexity k and no edge of complexity $k+1$, where the *complexity of an edge* is given by the number of segments and circular arcs needed to represent the edge. In this paper, we seek for smooth orthogonal layouts of low edge complexity.

1.1 Motivation

Recent work suggests attractive alternatives that address readability related issues posed by the presence of bends in polyline drawings. Such work is motivated by perception research, indicating that representing paths with smooth geodesic trajectories aids in comprehension [26], as well as by the aesthetic appeal of drawings with smooth curves such as those of American abstract artist Mark Lombardi [32]. Two features that stand out in Lombardi's work are the use of circular-arc edges and their even distribution around vertices; see Figure 2. Such even spacing of edges around each vertex (also known as *perfect angular resolution*), together with the use of circular arcs for edges, formally define *Lombardi drawings* of graphs [16, 17].

Not all graphs allow for Lombardi realizations and the characterization of Lombardi graphs is an open problem. One way to visualize non-Lombardi graphs in a Lombardi fashion is to relax the circular-arc constraint; while vertices still have perfect angular resolution, the edges can be represented as smooth sequences of circular arcs. For example, Duncan *et al.* [15] describe k-Lombardi drawings, where each edge is a smooth sequence of k circular arcs.

Note that vertices of degree four have perfect angular resolution in traditional orthogonal graph layouts, by virtue of construction, and vertices of lower degrees have angular resolution within a factor of two of optimal. In this paper, we study the problem of creating smooth orthogonal layouts, where we use circular arcs to create smoother curves for the edges in conjunction with the horizontal and vertical line segments of the edges. In order to obtain smooth curves, we ensure that each edge is composed of rectilinear line segments and circular arcs with common tangents.

Our general approach is based on modifying a given traditional orthogonal layout by moving the vertices as needed, and replacing each bend by a smooth circular arc of appropriate radius, without introducing edge-crossings. We show that in many settings this can be accomplished without increasing edge complexity. As a result, we hope that we will eventually obtain layouts in which it is easier to follow non straight-line edges, which are represented by smooth curves (as the human eye follows smooth curves; bends interrupt the eye movement and require changes of direction).

Figure 3 shows that the use of circular arcs can also reduce edge complexity. It is easy to see that any traditional orthogonal layout of K_3 has complexity at least two; see Figure 3a. Allowing circular arcs reduces the complexity to one; see Figure 3b. Similarly, the complexity-2 layout of the cube graph can be transformed into a smooth complexity-1 layout; see Figures 3c-3d. However, as a different layout of the cube graph demonstrates, we cannot always obtain a smooth layout of complexity one by simply replacing the segments adjacent to a bend by a circular arc; see Figure 3e.

1.2 Related Work

Early work on orthogonal layouts was done by Valiant [36] and Leiserson [30] in the context of VLSI design. Tamassia [34], Tamassia and Tollis [35], and Biedl and Kant [3] continued this line of research in the context of graph drawing. The common objectives have been the minimization of the used area, total edge length, total number of bends, and maximum number of bends per edge. By default it was often assumed that input graphs were restricted to degree-4 planar graphs. Models incorporating higher degree graphs were introduced later by Tamassia [34] and Fößmeier and Kaufmann [21].

Chernobelsky *et al.* [10] relax the perfect angular resolution constraint in Lombardi drawings and describe functional force-directed algorithms, which produce aesthetically appealing near-Lombardi drawings. In addition to the work on Lombardi drawings, there has been other work on graph drawing with circular-arc or curvilinear edges for the sake of achieving good angular resolution [9, 23]. There is also significant work on *confluent drawings* [14, 18, 19, 24, 25], where curvilinear edges are used not to separate edges, but rather to bundle similar edges together and avoid edge crossings. In confluent drawings, edges are drawn like train-tracks using locally-monotone curves which do not self-intersect and which do not have sharp turns. The curves may have overlapping portions, but no crossings.

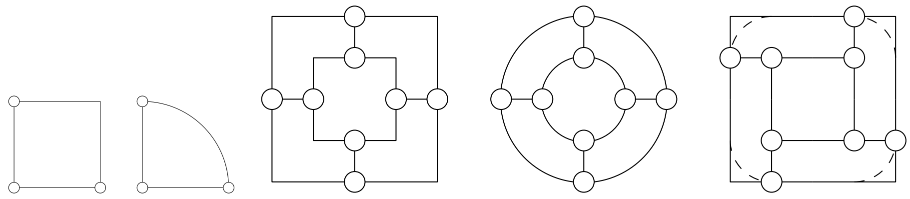


Figure 3: (a) An orthogonal layout of the triangle graph with edge complexity two. (b) A smooth layout of the triangle graph with edge complexity one. (c) An orthogonal layout of the cube graph with edge complexity two. (d) A smooth layout of the cube graph with edge complexity one. (e) An orthogonal layout of the cube which cannot be improved w.r.t. edge complexity.

Aichholzer *et al.* [1] show that, for a given embedded planar triangulation with fixed vertex positions, one can find a circular-arc drawing of the triangulation that maximizes the minimum angular resolution by solving a linear program. Brandes and Wagner [7] provide a force-directed method for visualizing train schedules using Bézier curves for edges and fixed positions for vertices. Finkel and Tamassia [20] extend this work with a force-directed method for drawing graphs with curvilinear edges where vertex positions are not fixed. For fixed position drawings with cubic Bézier curves, Brandes and Schlieper [5] use force-directed methods to maximize angular resolution and Brandes *et al.* [6] rotate optimal angular resolution templates.

1.3 Our Contribution

We are particularly interested in providing theoretical guarantees about creating smooth orthogonal layouts, while not increasing edge complexity and not introducing edge crossings. Initially, we focus on a quite restricted layout model, referred to as *fixed layout model*, according to which an orthogonal layout is given and the placement of the vertices cannot be changed (Section 3). We prove that we can minimize the number of segments of the given layout, by appropriately replacing each bend by a circular arc segment, in the case where all circular arc segments have the same radius (Theorem 1).

We next consider a more flexible model, referred to as *fixed shape model*, according to which an orthogonal layout is again given, but in this setting the shape of the layout (i.e., the port-assignment and the sequence of directional changes of the edges) should be preserved (Section 4). Among others, we prove that if the input graph has a complexity-2 traditional orthogonal layout, we can transform it into a smooth complexity-2 layout (Theorem 2). However, if the input graph has a complexity-3 traditional orthogonal layout, we can transform it into a smooth complexity-4 layout (Theorem 3), i.e., the edge complexity is increased. Theorem 4 and Theorem 5 suggest that if the input graph has either a complexity-3 traditional orthogonal layout in which all edges turn in the same direction, or, a bend-optimal layout, respectively, then it is possible to preserve the edge complexity in the output smooth orthogonal layout. However, in general our technique increases the layout area by a factor of n and the edge complexity by no more than a factor of $\lceil 3/2 \cdot k \rceil - 1$, where k is the complexity of the input orthogonal layout (Theorem 6). On the other hand, if one wants to reduce the edge complexity, the area penalty can be exponential, as shown in Theorem 7.

In Section 5, we study how much the complexity can be reduced if we are allowed to change the shape of the drawing. We first show that every 4-planar graph admits a smooth orthogonal layout with edge complexity 3 (Theorem 8). This is close to optimal, since there exists a graph that does not admit a complexity-1 smooth orthogonal layout (Theorem 9) and an infinite class of graphs whose members also do not admit complexity-1 smooth orthogonal layouts if the outerface is fixed (Theorem 10). We also show that any triconnected 3-planar or Hamiltonian 3-planar graph admits a smooth orthogonal layout with edge complexity 1 (Theorems 11 and 12, respectively). Using the Kandinsky model for removing the degree restriction (Section 6), we demonstrate that any planar graph has a smooth complexity-1 layout, if it is Hamiltonian (Theorem 13), or, a smooth complexity-2 layout in general (Theorem 14).

2 Preliminaries

Let $G = (V, E)$ be a simple undirected graph with n vertices, $n \geq 3$, and m edges. For a subset $V' \subset V$, we denote by $G[V']$ the subgraph of G induced by V'. By $deg_G(v)$ we denote the degree of vertex v in G. A graph is called *k-connected* if the removal of $k-1$ vertices does not disconnect the graph. Two vertices whose removal disconnects the graph are referred to as a *separation pair*. If G is planar and triconnected, then it has a unique planar embedding up to the choice of the outerface [11]. A path P is a sequence $\{z_0, z_1, \ldots, z_\ell\}$ of distinct adjacent vertices, i.e., $(z_i, z_{i+1}) \in E[G]$, $i = 0, \ldots, \ell - 1$.

Definition 1 (Canonical Ordering [12, 28]) *Let $\Pi = \{P_0, \ldots, P_s\}$ be a partition of V into paths and let $P_0 = \{v_1, v_2\}$, $P_s = \{v_n\}$ such that $\{v_2, v_1, v_n\}$ is a path on the outerface of G in clockwise direction. For $k = 0, \ldots s$, let $G_k = G[V_k] = (V_k, E_k)$ be the subgraph induced by $V_k = P_0 \cup \ldots \cup P_k$, let C_k be the outerface of G_k. Partition Π is a* canonical ordering *of (G, v_1) if for each $k = 1, \ldots, s$:*

i) C_k is a simple cycle.

ii) Each vertex z_i in P_k has a neighbor in $V - V_k$.

iii) $|P_k| = 1$ or $deg_{G_k}(z_i) = 2$ for each vertex z_i in P_k.

A canonical ordering Π is refined to a *canonical vertex ordering* $\{v_1, \ldots, v_n\}$ by ordering the vertices in each P_k according to their clockwise appearance on each C_k, $k > 0$. Note that a canonical ordering of (G, v_1) is not uniquely defined. Let $\{P_0, \ldots, P_k\}$ be a sequence of paths that can be extended to a canonical ordering of G. A path P of G is a *feasible candidate* for the step $k+1$ if $\{P_0, \ldots, P_k, P\}$ can be extended to a canonical ordering. Let $v_1 = c_1, c_2, \ldots, c_q = v_2$ be the vertices from left to right on C_k. Let c_ℓ (c_r) be the neighbor of P on C_k such that ℓ (r) is as small (large) as possible. We call c_ℓ (c_r) the *left* (*right*) *neighbor* of P.

Definition 2 (Leftmost Canonical Ordering [28]) *Let* $\Pi = \{P_0, \ldots, P_s\}$ *be a canonical ordering.* Π *is called* leftmost *if for* $k = 0, \ldots, s-1$ *the following is true. Let* c_ℓ *be the left neighbor of* P_{k+1} *and let* $P_{k'}$*,* $k+1 \leq k' \leq s$*, be a feasible candidate for the step* $k+1$ *with left neighbor* $c_{\ell'}$*. Then* $\ell \leq \ell'$*.*

3 Smooth Orthogonal Layouts under the Fixed Layout Model

The most restrictive version of our approach is the one where the layout of the graph is given and the placement of the vertices cannot be changed. In this setting, we are only allowed to replace the bends of the edges by circular arcs, such that adjacent segments have the same horizontal or vertical tangent at their contact points. This restriction is referred to as the *fixed layout model.* An ad-hoc approach would be to replace each bend by a very small circular arc segment, which would increase the number of segments on the edge from $k > 0$ to $2k-1$. Such increase in edge complexity might be unavoidable in the fixed layout setting; see Figure 4.

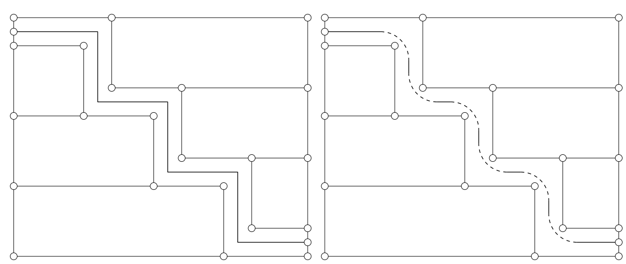


Figure 4: Edge complexity might increase from k to $2k-1$ for "staircase" edges.

In practice, it might be possible to avoid increasing the edge complexity. One way to achieve this in the fixed layout model is to try to increase the radii of the circular arcs until one of their adjacent straight-line segments disappear or the circular segment hits another graphical object that prevents a further enlargement. In fact, it is easy to minimize the number of segments in the fixed layout model, if all circular arcs have the same radius, as the next lemma shows.

Theorem 1 *Given an orthogonal layout, there exists an* $O(N \log N)$*-algorithm that maximizes the uniform radii of the circular arcs in the drawing under the* *fixed layout model, where N is the total number of vertices and bends of the orthogonal layout.*

Proof: The algorithm determines whether a smooth layout of an input radius is feasible (i.e., leads to a crossing-free solution) by a plane-sweep method. The status of the sweep line is the order of the edge segments intersecting it. The event points of the plane sweep algorithm are the vertices and the bends of the orthogonal layout [33]. Since we are not interested in reporting all possible intersections, the corresponding decision problem can be answered in $O(N \log N)$ time. Then, it is sufficient to apply the randomized optimization technique of Chan to solve the problem in $O(N \log N)$ time [8]. □

4 Smooth Orthogonal Layouts under the Fixed Shape Model

In this section, we assume that a layout is given, but now we are allowed to change the length of the segments of the edges, so long as no segments become zero length. Specifically, in this setting the "shape" of the layout is fixed, i.e., if an edge connects to a vertex using the north port, then it must continue to use the north port and the sequence of directional changes of an edge cannot be modified. This restriction is referred to as the *preserved orthogonal representation* in [34]. Here, we call it the *fixed-shape model*. Even though this model is also very restrictive, it provides us with enough flexibility to produce smooth layouts with low edge complexity.

4.1 Smooth Postprocessing: Layout Stretching

Traditional orthogonal layout algorithms place the vertices of an input 4-planar graph on an integer grid of size $O(n) \times O(n)$. In the following, we describe a technique which transforms (under the fixed shape model) an orthogonal layout of a certain edge complexity, into a smooth orthogonal layout with comparable edge complexity but increased layout area by a factor of n (i.e., from $O(n) \times O(n)$ to $O(n^2) \times O(n)$). Our technique can be considered as a postprocessing of a traditional orthogonal layout, in which the goal is to eventually obtain a smooth orthogonal layout of similar shape.

We begin with a simple problem, that of postprocessing a traditional orthogonal layout in which all edges have complexity 2, so as to obtain a smooth layout of the same complexity. In this scenario, we use circular arcs of varying sizes to replace straight-line segments (unlike in Theorem 1, where we used circular arcs of the same size).

Theorem 2 *Let G be an n-vertex 4-planar graph that admits a complexity-2 orthogonal layout Γ in $O(n) \times O(n)$ area. There exists an $O(n)$-time algorithm that transforms Γ into a complexity-2 smooth orthogonal layout of G in $O(n^2) \times O(n)$ area, under the fixed shape model.*

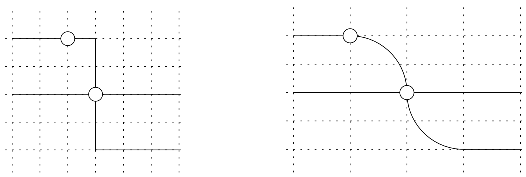


Figure 5: Illustration of horizontal stretching.

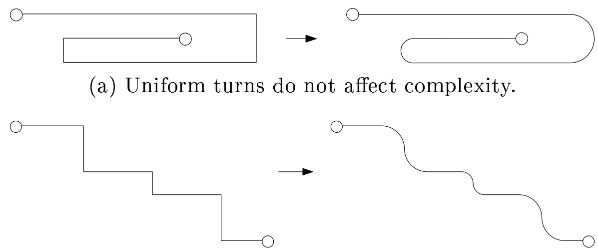
Proof: Let l be the length of the longest vertical segment in any complexity-2 edge of the input layout. Consider the vertex v which is one of the endpoints of that vertical segment of length l. Stretch the entire drawing horizontally by a factor of l; see Figure 5. Then, in the stretched drawing, the vertical segment is no larger than its matching horizontal segment. Due to the stretching, the grid of size $l \times l$ in each quadrant of vertex v is empty from other vertices and edges. Hence, we can safely replace the vertical segment with a quarter-circle arc, which yields a smooth complexity-2 realization of the edge.

Note that the same argument can be applied to any complexity-2 edge in the original layout. That is, for any such edge the vertical segment is no larger than the horizontal segment, and there is an empty square grid in each quadrant around the vertex, allowing us to replace the vertical segment with a circular arc. This immediately implies that once this procedure has been used to modify all complexity-2 edges, the result is a smooth complexity-2 orthogonal layout on a grid that is a factor of n larger than the input layout, since in worst case $l = O(n)$. The safe insertion of circular arcs in place of straight-line segments ensures that if we started without crossings, we also finish without crossings. Since the stretching was applied only once, the transformation can be accomplished in linear time using one plane sweep to stretch the drawing and another one to introduce circular arcs of appropriate sizes. □

Theorem 3 *Let G be an n-vertex 4-planar graph that admits a complexity-3 orthogonal layout Γ in $O(n) \times O(n)$ area. There exists an $O(n)$-time algorithm that transforms Γ into a complexity-4 smooth orthogonal layout of G in $O(n^2) \times O(n)$ area, under the fixed shape model.*

Proof: We utilize the stretching technique described in the proof of Theorem 2. Then, for complexity-1 or complexity-2 edges of the input orthogonal layout Γ, the edge complexity does not increase. This also holds for edges that turn only in the same direction (i.e., right-right, or, left-left); see Figure 6a. However, for edges that turn in alternating directions (i.e., left-right, or, right-left), the edge complexity increases from 3 to 4; see Figure 6b. Hence, Γ is eventually transformed into a complexity-4 smooth orthogonal layout. □

Edges of complexity-3 that turn in alternating directions are usually called *S-shaped edges* or *zig-zags*. From the proof of Theorem 3, it follows that if the input orthogonal layout contains no S-shaped edges, then it can be transformed into a smooth orthogonal layout of edge complexity-3, which implies that edge



(a) Uniform turns do not affect complexity.

(b) Alternating turns increase complexity.

Figure 6: Illustration of horizontal stretching for edges with many bends.

complexity of the input orthogonal layout does not increase. This observation is summarized in the following theorem.

Theorem 4 *Let G be an n-vertex 4-planar graph that admits a complexity-3 orthogonal layout Γ in $O(n) \times O(n)$ area that contains no S-shaped edges. There exists an $O(n)$-time algorithm that transforms Γ into a complexity-3 smooth orthogonal layout of G in $O(n^2) \times O(n)$ area, under the fixed shape model.*

Theorem 5 *Let G be an n-vertex 4-planar graph for which an orthogonal layout Γ with the minimum number of bends has complexity k, where $k > 1$. There exists an $O(n)$-time algorithm that transforms Γ into a complexity-k smooth orthogonal layout of G in $O(n^2) \times O(n)$ area, under the fixed shape model.*

Proof: Since Γ is of minimum number of bends, it contains no S-shaped edges [34], or equivalently, all edges of complexity more than one turn in the same direction (e.g., right-right-right). This implies that if we utilize the stretching technique described in the proof of Theorem 2, the result is a smooth orthogonal layout of G, which has the same edge complexity as Γ, but it is a factor of n larger than the input layout. Now observe that Γ has $O(n)$ bends, thereby $O(n)$ edge segments. This suggest that (after deleting potential empty rows/columns), Γ needs $O(n) \times O(n)$ area, establishing the bound of $O(n^2) \times O(n)$ of the theorem. □

Theorem 6 *Let G be an n-vertex 4-planar graph that admits a complexity-k orthogonal layout Γ in $O(n) \times O(n)$ area, where $k > 1$. There exists an $O(n)$-time algorithm that transforms Γ into a complexity-($\lceil 3/2 \cdot k \rceil - 1$) smooth orthogonal layout of G in $O(n^2) \times O(n)$ area, under the fixed shape model.*

Proof: Again, we utilize the stretching technique described in the proof of Theorem 2. As already stated, for complexity-1 or complexity-2 edges or edges that turn only in the same direction in the input orthogonal layout Γ, the edge complexity does not increase. However, for edges that turn in alternating directions (i.e., staircase edges), the edge complexity increases from k to $\lceil 3/2 \cdot k \rceil - 1$. To realize this, observe that the number of horizontal segments of the stretched layout equals to the number of horizontal segments of Γ (i.e., $\lceil k/2 \rceil$), while the number of bends of Γ (which are $k-1$ in total) are in one to one correspondence with the circular arc-segments of the stretched layout. This implies that stretching increases edge complexity by no more than a factor of $\lceil 3/2 \cdot k \rceil - 1$. □

4.2 Area Bounds

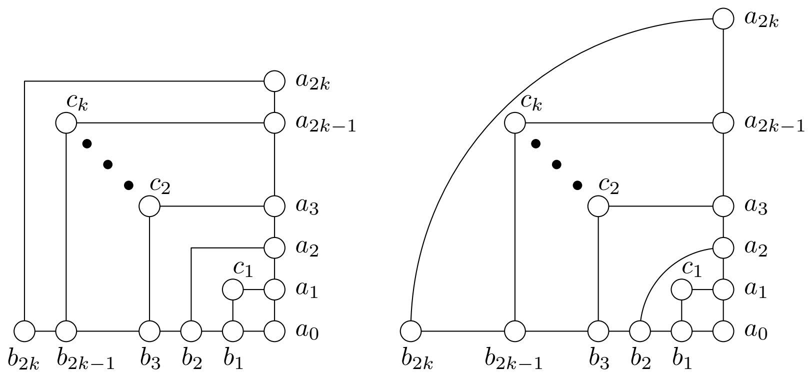
Our technique for creating smooth orthogonal layouts under the fixed shape model results in increased drawing area. In particular, when the stretching technique described in Section 4.1 is applied to an orthogonal layout of a certain edge complexity, the result is a smooth orthogonal layout that requires increased layout area from $O(n) \times O(n)$ to $O(n^2) \times O(n)$.

The situation is completely different, if one wants to generate smooth orthogonal layouts with complexity exactly one, under the fixed shape model. In particular, for smooth complexity-1 layouts, the area penalty can be exponential, as shown in the next theorem.

Theorem 7 *There exists an n-vertex 4-planar graph G that admits a complexity-2 orthogonal layout Γ in $O(n) \times O(n)$ area, whose corresponding smooth complexity-1 layout requires exponential area, under the fixed shape model.*

Proof: We show this claim for $n = 5k + 1$, for some integer $k \geq 1$; for all other values of n we can create such a graph by adding a few degree-1 vertices to graph G. Graph $G = (V, E)$ and its corresponding complexity-2 orthogonal layout Γ are illustrated in Figure 7a. Observe that the vertex set V of G is partitioned into three disjoint sets: $V_a = \{a_0, \ldots, a_{2k}\}$, $V_b = \{b_1, \ldots, b_{2k}\}$ and $V_c = \{c_1, \ldots, c_k\}$, such that consecutive vertices in V_a (V_b, resp.) form a path that is drawn along a vertical (horizontal, resp.) line in Γ. In the drawing Γ of G, vertex a_0 is drawn at the intersection of these two lines, while consecutive vertices of the two paths are separated by exactly one unit of length. For each $i = 1, 2, \ldots, 2k$, if i is even then $(a_i, b_i) \in E$ is drawn with one bend, otherwise $(a_i, c_{(i+1)/2}) \in E$ and $(a_i, c_{(i+1)/2}) \in E$ are both drawn without bends. Clearly, the area occupied by drawing Γ equals to $(2k+1) \times (2k+1)$, which is quadratic to the number of vertices of G, since $k = O(n)$.

Now recall that one is not allowed to change the port assignment in the fixed shape model. Hence, in a smooth orthogonal layout of G with edge complexity-1, each bent edge of Γ should be replaced by a quarter-circle arc, as illustrated in Figure 7b. The shape of the remaining edges of G should not be affected. For a vertex $v \in V$, we denote by $d(v, a_0)$ the distance of v from vertex a_0 in the smooth orthogonal layout derived from Γ. Clearly, $d(a_0, a_0) = 0$. For $i = 1, 2, \ldots, k-1$, it follows from the planarity of the derived layout that $d(a_{2i+1}, a_0) = d(b_{2i+1}, a_0) \geq d(a_{2i}, a_0) + 1$, and, $d(a_{2i+2}, a_0) = d(b_{2i+2}, a_0) \geq \lceil \sqrt{2} \cdot d(a_{2i+1}, a_0) \rceil$, which implies that $d(a_{2k}, a_0) = d(b_{2k}, a_0) = O(\sqrt{2}^{\frac{n}{2}})$. This completes the proof of the exponential area requirement. □



(a) An orthogonal complexity-2 layout which fits in quadratic area.

(b) Exponential area is required for the corresponding smooth complexity-1 layout.

Figure 7: Illustration of exponential area blow-up for complexity-1 smooth orthogonal layouts.

5 General Smooth Layouts with Low Complexity

While many 4-planar graphs indeed have complexity-2 orthogonal layouts, and can hence be transformed into smooth complexity-2 layouts (with the aid of Theorem 2), this is not true for all graphs. What is known, is that every 4-planar graph, except the octahedron, has a complexity-3 orthogonal drawing. The octahedron is a special case of a 4-regular graph which requires complexity 4 (i.e., 3 bends per edge); see Figure 1c. In the following we show that all 4-planar graphs (including the octahedron) admit smooth complexity-3 layouts. This is next to optimal, as we also show that for smooth layouts, complexity 2 is necessary.

Theorem 8 *Let G be an n-vertex 4-planar graph. There exists an $O(n)$-time algorithm that computes a complexity-3 smooth orthogonal layout of G in $O(n^2) \times O(n)$ area.*

Proof: As already stated, any 4-planar graph, except the octahedron, admits an orthogonal layout of edge complexity 3. This is due to a linear-time constructive algorithm of Biedl and Kant [3]. Hence, from Theorem 3 immediately follows that a complexity-4 smooth orthogonal layout of G in $O(n^2) \times O(n)$ area exists. Now observe that the main difficulty in getting a better bound are S-shaped edges (recall Theorem 2). But S-shaped edges can always be eliminated. This was explicitly shown by Liu, Morgana and Simeone [31], who presented a linear-time algorithm to find an orthogonal layout of a given 4-planar graph in $O(n) \times O(n)$ area, in which (i) each edge is guaranteed to have complexity at most 3, and, (ii) S-shaped edges do not exist, i.e., all edges turn in the same direction (again the octahedron is the only exception, as complexity-4 is required). Since the octahedron admits a complexity-2 smooth orthogonal layout (see Figure 1g), by Theorem 2 it follows that any 4-planar graph admits a complexity-3 smooth orthogonal layout in $O(n^2) \times O(n)$ area, which can be computed in $O(n)$-time. □

The use of circular arcs allows us not only to create smooth orthogonal layouts without increasing edge complexity, but it sometimes allows us to reduce the edge complexity. For example, we can compute smooth orthogonal layouts with reduced complexity for all 4-planar Platonic solids. The tetrahedron, cube, and dodecahedron, which require complexity-2 in traditional orthogonal layouts, all have smooth complexity-1 layouts; see Figure 1. However, we cannot always achieve this, as shown in the next theorem.

Theorem 9 *There exists a* 4*-planar graph that does not admit a complexity-*1 *smooth orthogonal layout.*

Proof: Consider the octahedron graph; it is not difficult to construct a smooth layout of complexity 2; see Figure 1g. To show that the graph does not have a smooth complexity-1 layout we use a 2-part geometric argument. First we show that there is only one way (up to rotation and scaling) to place the three vertices on the outerface. Then we show that given the placement of the outerface, there is no valid placement for the internal vertices.

Consider the octahedron graph and suppose that it has a smooth complexity-1 layout. As the graph is 4-regular and very symmetric, we can take any face as the outerface. The outerface is formed by three vertices of degree four, and its edges must be arranged in such a way that each vertex has two free ports pointing inside. Given these conditions it is easy to show by examining all different realizations of the triangle graph that neither of three edges on the outerface can be a straight-line segment or a quarter-circle arc (see Figure 8a). In fact, the only way to realize the face and keep the ports inside is with two half-circle arcs and one 3/4-circle arc (see Figure 8b). Moreover, this feasible configuration is unique, up to rotation and scaling.

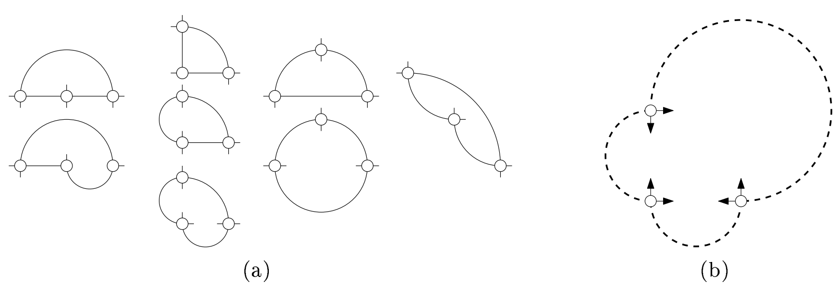


Figure 8: (a) Different realizations of the triangle graph with edge complexity one, in which quarter-circle arcs can be appropriately replaced by 3/4-circle arcs to obtain all different realizations of the triangle graph. (b) A realization with two half-circle arcs and one 3/4-circle arc in which the ports are kept inside.

Note that the only feasible realization of the outerface places the three vertices at the corners of a square (a consequence of the use of two half-circles and one 3/4-circle). Now we must place the inner three vertices of the octahedron. Consider one of the two inner vertices that is adjacent to two outer vertices that are connected by a half-circle. The inner vertex must use two consecutive ports as connections to the outerface and leave two free ports pointing inside. Using straight-forward case analysis, it is not difficult to show that there is no feasible placement for such a vertex. □

Note that the results for smooth complexity in many ways mirror the results for the complexity of orthogonal drawings: (i) In traditional orthogonal graph drawing, any 4-planar graph (except the octahedron) can be drawn with edge complexity 3, if one is allowed to choose the outerface and the planar embedding; in smooth orthogonal graph drawing, any 4-planar graph admit a complexity-3 smooth orthogonal drawing, if one is allowed to choose the outerface and the planar embedding (Theorem 8). (ii) In traditional orthogonal graph drawing, any 4-planar graph for which the outerface is fixed to be a triangle of degree-4 vertices requires an edge with complexity 4; in smooth orthogonal graph drawing, any 4-planar graph for which the outerface is fixed to be a triangle of degree-4 vertices requires an edge with complexity 2. We prove this now:

Theorem 10 *There exist infinitely many* 4*-planar graphs that do not admit smooth complexity-*1 *layouts, if the outerface is fixed.*

Proof: We construct a class of 4-planar graphs with a fixed outerface consisting of three vertices of degree four and a cycle with k vertices inside, where $k \geq 3$. The cycle has three special vertices, each of which is connected to a pair of vertices of the outerface. Note that the case where $k = 3$ corresponds to the octahedron discussed above. In the case where $k > 3$, the three special vertices of the cycle also have degree 4 and must be connected to a pair of vertices incident to the outerface. Each of these special vertices must use consecutive ports to connect to the outerface and must leave two adjacent ports pointing inside available for their neighbors on the cycle. Just as in the case of the octahedron, it is impossible to place all three of the special vertices inside the outerface under these constraints, and use only complexity-1 edges. □

In the following we turn our attention on subclasses of 4-planar graphs, for which we can prove that admit complexity-1 smooth orthogonal layouts. In particular, we prove that all triconnected 3-planar graphs and all Hamiltonian (not necessarily triconnected) 3-planar graphs admit complexity-1 smooth orthogonal layouts.

Theorem 11 *Let G be an n-vertex* 3*-planar triconnected graph. There exists an $O(n)$-time algorithm that computes a complexity-*1 *smooth orthogonal layout of G in $O(n) \times O(n)$ area.*

Proof: Kant [27] proved that every 3-planar triconnected graph on n vertices admits a layout on an $n \times n/2$ hexagonal grid, in which all edges except one are drawn either as rectilinear or as diagonal segments. In a high level description, his algorithm utilizes a leftmost canonical order Π (refer to Section 2) of the input graph G to draw it. In particular, the algorithm processes each path of Π in turn. Assuming that zero or more paths of Π have been processed and $P = \{z_1, z_2, \ldots, z_\lambda\} \in \Pi$ is the next path of Π to be processed, then all of its vertices are drawn in unit-distanced points along the next unoccupied horizontal grid-line on top of the drawing constructed so far, such that (c_ℓ, z_1) ((c_r, z_λ), resp.) is drawn as a diagonal (vertical, resp.) line-segment of positive slope, where c_ℓ and c_r are the left and right neighbors of P. For example, in Figure 9a vertices 15 and 4 are the left and right neighbors of path $\{16, 17\}$ of the canonical order, respectively. Hence, edges $(15, 16)$, $(16, 17)$ and $(17, 4)$ are drawn as diagonal, horizontal and vertical line-segments, respectively. Note that in the drawing the bent edge is the one that connects the first with the last vertex of the canonical order.

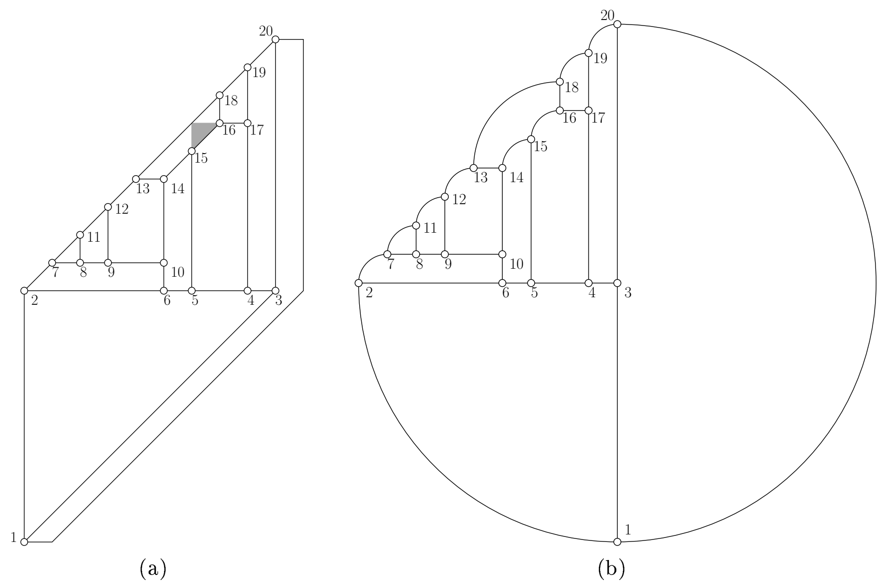


Figure 9: (a) Orthogonal layout with exactly one bent edge [27]. (b) A smooth complexity-1 orthogonal layout derived from the one of Figure 9a.

In order to obtain a smooth orthogonal layout of complexity-1 from the output orthogonal layout of the algorithm of Kant, we utilize the following property: If an edge $e = (u, v)$ is drawn as a diagonal segment of positive slope in the orthogonal layout with endpoints $p_u = (x_u, y_u)$ and $p_v = (x_v, y_v)$, such that $x_u > x_v$ and $y_u > y_v$, then the triangle formed by p_u, p_v and (x_v, y_u) contains no vertices of the graph (for an example refer to the gray colored triangle of Figure 9a). Hence, it is safe to replace the diagonal segment connecting p_u and

p_v with a quarter circular arc which utilizes the left port of u and the top port of v. The bent edge can also be drawn with smooth complexity-1, if the first vertex of the canonical order is appropriately placed such that its x-coordinate is the same as the one of the last vertex of the canonical order; see Figure 9b. Asymptotically, this does not affect the area of the drawing, which remains quadratic. □

Theorem 12 *Let G be an n-vertex 3-planar Hamiltonian graph and C_G be a given Hamiltonian cycle of G. There exists an $O(n)$-time algorithm that computes a complexity-1 smooth orthogonal layout of G in $O(n) \times O(n)$ area.*

Proof: We draw the vertices of G in unit-distanced points along a horizontal line, say ℓ, from left to right in the order that appear, when traversing C_G in clockwise direction starting from an arbitrarily selected vertex of it. Vertices adjacent in C_G are connected with edges drawn as horizontal line-segments, except for the one that connects the leftmost with the rightmost vertex of G along ℓ which is drawn as half-circle on the top half-plane. Now observe that C_G splits the remaining edges of the graph into two groups: one with edges inside C_G and another with edges outside C_G. We route these edges using half-circles above and below ℓ, respectively. Since the maximum degree of G is 3, we guarantee that no two edges will use the same top or bottom port. In addition, since the planar embedding is maintained, no crossings are introduced. To complete the proof, it is easy to see that the area occupied by the drawing is $n \times n$, i.e., quadratic. Note that the drawing resembles a book embedding. □

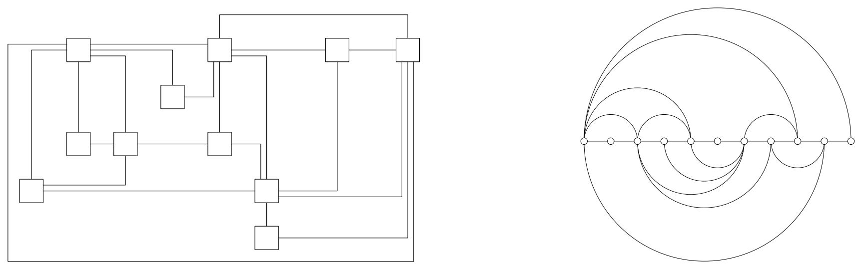
6 Smooth Layouts for High Degree Graphs

A serious limitation for the practical applicability of orthogonal layouts in general, and consequently for smooth orthogonal layouts, is the vertex degree restriction. Several extensions that overcome this restriction have been proposed for orthogonal layouts [34]. A quite common approach is the *Podevsnef* model [21], also known as Kandinsky model [4], where the basic idea is to use square-shaped nodes, placed on a coarse grid, with multiple edges attached to each side of the square aligned on finer grid; see Figure 10a.

We will apply our approach for making orthogonal layouts smooth to the Kandinsky model, requiring that different edges at the same side of a node must be circular-arcs of different radii. Of course, if the input graph is Hamiltonian, then similarly to Theorem 12 we can prove the following theorem; for an example refer to Figure 10b.

Theorem 13 *Let G be an n-vertex planar Hamiltonian graph and C_G be a given Hamiltonian cycle of G. There exists an $O(n)$-time algorithm that computes a complexity-1 smooth orthogonal layout of G in $O(n) \times O(n)$ area, under the Kandinsky model.*

For non-Hamiltonian graphs, a layout algorithm suitable for our model is the one of Kaufmann and Wiese [29] for point-set embedding with few bends



(a) (b)

Figure 10: (a) A Podevsnef style layout [21]; (b) A smooth Kandinsky layout of the same graph.

per edge[1]. This particular algorithm splits separating triangles by inserting dummy vertices on appropriate edges and triangulates again to make the graph 4-connected. In this way, the general case can be reduced to the Hamiltonian case, but an edge may now contain a middle dummy vertex on the horizontal line, and hence it might consist of two segments, an upper and a lower half-circle. It is easy to see that this results in a smooth layout (that resembles book embedding) with edges of complexity at most two, which yields the theorem below.

Theorem 14 *Let G be an n-vertex planar graph. There exists an $O(n^2)$-time algorithm that computes a complexity-2 smooth orthogonal layout of G in $O(n) \times O(n)$ area, under the Kandinsky model.*

7 Conclusion and Open Problems

In this paper, we introduced and presented the first combinatorial results for smooth orthogonal layouts, which follow the paradigms of traditional orthogonal layouts, but replace right-angle turns with circular arcs. We measured the edge complexity by the maximum number of straight or circular segments needed for any edge in the layout. We showed that any complexity-2 orthogonal layout can be transformed into a smooth complexity-2 layout in linear time. We also showed that every 4-planar graph has a smooth complexity-3 layout. In both cases, the price for smooth edges was a linear blowup in drawing area. That is, while the traditional orthogonal layout can fit in an $O(n) \times O(n)$ area, the smooth layout requires $O(n^2) \times O(n)$ area, where n is the number of vertices of the graph. Using the Kandinsky model for removing the degree restriction, we showed that any planar graph admits a smooth complexity-2 layout. Further, we showed that while in some cases the use of circular arcs can lower complexity, there are graphs that do not have smooth complexity-1 drawings, if the outerface is fixed.

[1]Alternatively, one could also use the layout algorithm of Di Giacomo *et al.* [13].

Of course, there are several natural open problems. In traditional orthogonal layouts the problem of testing whether a given graph has an embedding with only straight-line segments is NP-hard. The complexity of the corresponding problem for smooth complexity-1 layouts is not known. Another question is whether all graphs that have complexity-2 orthogonal layouts also admit smooth complexity-1 layouts. More generally, does there exist a 4-planar graph that has a complexity-k layout but does not have a smooth complexity-$(k-1)$ layout for any $k > 1$? Under the Kandinsky model, we proved that all planar graphs admit complexity-2 smooth orthogonal layouts. So, a natural question that arises in this context is whether this bound is tight, i.e., do all planar graphs admit complexity-1 smooth orthogonal layouts?

Many graph drawing tools produce orthogonal drawings without any guarantees on the number of bends per edge. In the context of developing post-processing methods for such tools, it would be desirable to obtain an algorithm that transforms a complexity-k orthogonal drawing into a smooth complexity-k orthogonal drawing. Note that while we have shown this is true for $k = 2$, our approach does not generalize to $k > 2$.

Acknowledgments

The authors would like to thank the anonymous reviewers for comments that helped in improving the readability of the paper and strengthening the results. We would also like to thank Martin Nöllenburg, Ignaz Rutter and Robert Krug for helpful discussions.

References

[1] O. Aichholzer, W. Aigner, F. Aurenhammer, K. Č. Dobiášová, and B. Jüttler. Arc triangulations. In *Proc. of 26th European Workshop on Computational Geometry*, pages 17–20, 2010. doi:10.1007/978-3-642-25878-7_29.

[2] S. N. Bhatt and F. T. Leighton. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences*, 28(2):300–343, 1984. doi:10.1016/0022-0000(84)90071-0.

[3] T. C. Biedl and G. Kant. A better heuristic for orthogonal graph drawings. In *Proc. of 2nd European Symposium on Algorithms*, volume 855 of *LNCS*, pages 24–35, 1994. doi:10.1007/BFb0049394.

[4] T. C. Biedl and M. Kaufmann. Area-efficient static and incremental graph drawings. In *Proc. of 5th European Symposium on Algorithms*, volume 1284 of *LNCS*, pages 37–52, 1997. doi:10.1007/3-540-63397-9_4.

[5] U. Brandes and B. Schlieper. Angle and distance constraints on tree drawings. In *Proc. of 14th International Symposium on Graph Drawing*, volume 4372 of *LNCS*, pages 54–65, 2007. doi:10.1007/978-3-540-70904-6_7.

[6] U. Brandes, G. Shubina, and R. Tamassia. Improving angular resolution in visualizations of geographic networks. In *Proc. of 2nd Eurographics Symposium on Visualization*, pages 23–32, 2000. doi:10.1007/978-3-7091-6783-0_3.

[7] U. Brandes and D. Wagner. Using graph layout to visualize train interconnection data. *Journal of Graph Algorithms and Applications*, 4(3):135–155, 2000. doi:10.7155/jgaa.00028.

[8] T. M. Chan. Geometric applications of a randomized optimization technique. *Discrete & Computational Geometry*, 22(4):547–567, 1999. doi:10.1007/PL00009478.

[9] C. C. Cheng, C. A. Duncan, M. T. Goodrich, and S. G. Kobourov. Drawing planar graphs with circular arcs. *Discrete & Computational Geom.*, 25(3):405–418, 2001. doi:10.1007/s004540010080.

[10] R. Chernobelskiy, K. Cunningham, M. Goodrich, S. G. Kobourov, and L. Trott. Force-Directed Lombardi-Style Graph Drawing. In *Proc. of 19th International Symposium on Graph Drawing*, volume 7034 of *LNCS*, pages 310–321, 2011. doi:10.1007/978-3-642-25878-7_31.

[11] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa. A linear algorithm for embedding planar graphs using PQ-trees. *Journal of Computer and System Sciences*, 30(1):54–76, 1985. doi:10.1016/0022-0000(85)90004-2.

[12] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. doi:10.1007/BF02122694.

[13] E. Di Giacomo, W. Didimo, G. Liotta, and S. K. Wismath. Curve-constrained drawings of planar graphs. *Computational Geometry*, 30(1):1–23, 2005. doi:10.1016/j.comgeo.2004.04.002.

[14] M. Dickerson, D. Eppstein, M. T. Goodrich, and J. Y. Meng. Confluent drawings: Visualizing non-planar diagrams in a planar way. *Journal of Graph Algorithms and Applications*, 9(1):31–52, 2005. doi:10.7155/jgaa.00099.

[15] C. A. Duncan, D. Eppstein, M. T. Goodrich, S. G. Kobourov, and M. Löffler. Planar and Poly-Arc Lombardi Drawings. In *Proc. of 19th International Symposium on Graph Drawing*, volume 7034 of *LNCS*, pages 309–319, 2011. doi:10.1007/978-3-642-25878-7_30.

[16] C. A. Duncan, D. Eppstein, M. T. Goodrich, S. G. Kobourov, and M. Nöllenburg. Drawing trees with perfect angular resolution and polynomial area. In *Proc. of 18th International Symposium on Graph Drawing*, volume 6502 of *LNCS*, pages 183–194, 2010. doi:10.1007/978-3-642-18469-7_17.

[17] C. A. Duncan, D. Eppstein, M. T. Goodrich, S. G. Kobourov, and M. Nöllenburg. Lombardi Drawings of Graphs. *Journal of Graph Algorithms and Applications*, 16(1):85–108, 2011. doi:10.7155/jgaa.00251.

[18] D. Eppstein, M. T. Goodrich, and J. Y. Meng. Delta-confluent drawings. In *Proc. of 13th International Symposium on Graph Drawing*, volume 3843 of *LNCS*, pages 165–176, 2005. doi:10.1007/11618058_16.

[19] D. Eppstein, M. T. Goodrich, and J. Y. Meng. Confluent layered drawings. *Algorithmica*, 47(4):439–452, 2007. doi:10.1007/s00453-006-0159-8.

[20] B. Finkel and R. Tamassia. Curvilinear Graph Drawing Using the Force-Directed Method. In *Proc. of 12th International Symposium on Graph Drawing*, volume 3383 of *LNCS*, pages 448–453, 2005. doi:10.1007/978-3-540-31843-9_46.

[21] U. Fößmeier and M. Kaufmann. Drawing high degree graphs with low bend numbers. In *Proc. of 3rd International Symposium on Graph Drawing*, volume 1027 of *LNCS*, pages 254–266, 1995. doi:10.1007/BFb0021809.

[22] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. In *Proc. of 3rd International Symposium on Graph Drawing*, volume 894 of *LNCS*, pages 286–297, 1995. doi:10.1007/3-540-58950-3_384.

[23] M. T. Goodrich and C. G. Wagner. A framework for drawing planar graphs with curves and polylines. *Journal of Algorithms*, 37(2):399–421, 2000. doi:10.1006/jagm.2000.1115.

[24] M. Hirsch, H. Meijer, and D. Rappaport. Biclique edge cover graphs and confluent drawings. In *Graph Drawing*, volume 4372 of *LNCS*, pages 405–416. 2007. doi:10.1007/978-3-540-70904-6_39.

[25] D. Holten and J. J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009. doi:10.1111/j.1467-8659.2009.01450.x.

[26] W. Huang, P. Eades, and S.-H. Hong. A graph reading behavior: Geodesic-path tendency. In *Proc. of IEEE Pacific Visualization Symposium (PacificVis09)*, IEEE, pages 137–144, 2009. doi:10.1109/PACIFICVIS.2009.4906848.

[27] G. Kant. Hexagonal grid drawings. volume 657 of *LNCS*, pages 263–276, 1993. doi:10.1007/3-540-56402-0_53.

[28] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16(1):4–32, 1996. doi:10.1007/BF02086606.

[29] M. Kaufmann and R. Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *Journal of Graph Algorithms Applications*, 6(1):115–129, 2002. doi:10.7155/jgaa.00046.

[30] C. E. Leiserson. Area-efficient graph layouts (for VLSI). In *Proc. of 21st Annual IEEE Symposium on Foundations of Computer Science*, volume 1547 of *IEEE*, pages 270–281, 1980. doi:10.1109/SFCS.1980.13.

[31] 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. doi:10.1016/S0166-218X(97)00076-0.

[32] M. Lombardi and R. Hobbs. *Mark Lombardi: Global Networks.* Independent Curators, 2003.

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

[34] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal of Computing*, 16(3):421–444, 1987. doi:10.1137/0216030.

[35] R. Tamassia and I. Tollis. Planar grid embedding in linear time. *IEEE Transactions on Circuits and Systems*, 36(9):1230–1234, 1989. doi:10.1109/31.34669.

[36] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Transaction on Computers*, 30(2):135–140, 1981. doi:10.1109/TC.1981.6312176.