

Smooth Orthogonal Layouts^{*}

Michael A. Bekos¹, Michael Kaufmann¹,
Stephen G. Kobourov², and Antonios Symvonis³

¹ Institute for Informatics, University of Tübingen, Germany

{bekos,mk}@informatik.uni-tuebingen.de

² Department of Computer Science, University of Arizona, USA

kobourov@cs.arizona.edu

³ School of Applied Mathematics and Physical Sciences, NTUA, Greece

symvonis@math.ntua.gr

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 biconnected 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.

1 Introduction

Orthogonal graph drawing has a long tradition, dating back to VLSI layouts and floor-planning applications. The input graphs are assumed to be planar and with max-degree four, also known as *4-planar graphs*. 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. Finding an embedding with the minimum number of bends is an NP-hard problem [11]; 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

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 drawings with improved readability and/or more aesthetic appeal; see Fig. 1. A natural notion of *edge complexity* for such curves is the combinatorial complexity,

* 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. The work of S. G. Kobourov is supported in part by NSF grant CCF-1115971 and a grant from the Humboldt Foundation.

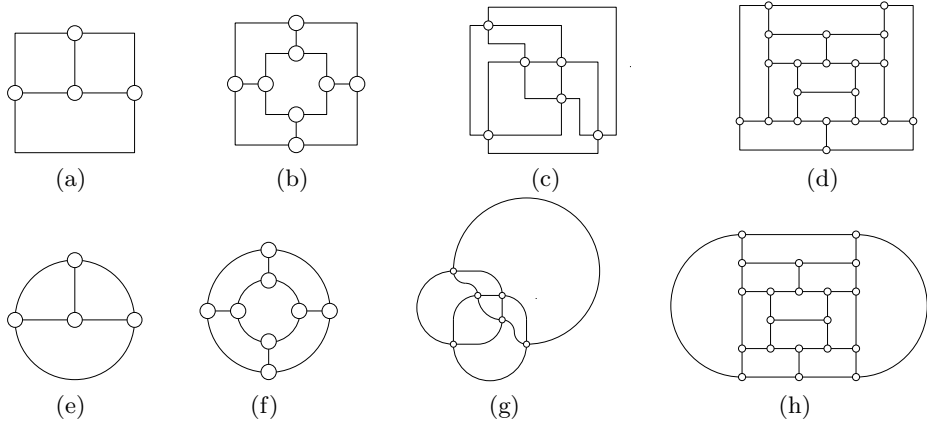


Fig. 1. All Platonic solids with degree 3 or 4 drawn in traditional orthogonal style with the minimum number of bends per edge and redrawn in the smooth orthogonal style with better edge complexity

given by the number of arcs forming an edge, and a natural optimization goal is to minimize the 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 [14], as well as by the aesthetic appeal of drawings such as those of American abstract artist Mark Lombardi [18]. Two features that stand out in Lombardi's work are the use of circular-arc edges and their even distribution around vertices. 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* [8].

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.* [7] describe *k*-Lombardi drawings, where each edge is a smooth sequence of *k* circular arcs.

Vertices of degree four have perfect angular resolution in traditional orthogonal layouts, by virtue of construction. 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 order to obtain smooth curves, we ensure that each edge is composed of rectilinear 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

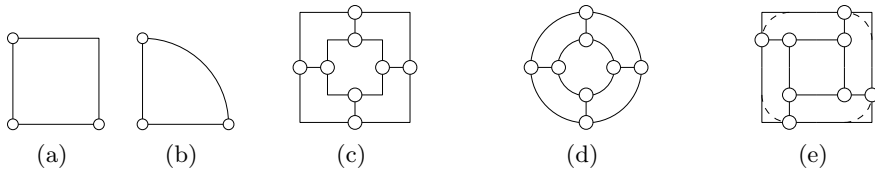


Fig. 2. (a) An orthogonal layout of the triangle graph with edge complexity two. (b) A smooth layout of the triangle graph with edge complexity one. (c) An orthogonal layout of the cube graph with edge complexity two. (d) A smooth layout of the cube graph with edge complexity one. (e) An orthogonal layout of the cube which cannot be improved w.r.t. edge complexity.

many settings this can be accomplished without increasing edge complexity. As a result, we obtain layouts in which it is easier to follow non straight-line edges.

The use of circular arcs can also reduce edge complexity. Any traditional orthogonal layout of K_3 has complexity at least two; see Fig. 2a. Allowing circular arcs reduces the complexity to one; see Fig. 2b. Similarly, the complexity-2 layout of the cube graph can be transformed into a smooth complexity-1 layout; see Fig. 2c-2d. 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 Fig. 2e.

1.2 Related Work

Early work on orthogonal layouts was done by Valiant [21] and Leiserson [17] in the context of VLSI design. Tamassia [19], Tamassia and Tollis [20], and Biedl and Kant [1] 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.

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 [6,12]. There is also significant work on *confluent drawings* [13], where curvilinear edges are used 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.

Brandes and Wagner [4] provide a force-directed method for visualizing train schedules using Bézier curves for edges and fixed positions for vertices. Finkel and Tamassia [9] 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 [3] use force-directed methods to maximize angular resolution.

2 Smooth Fixed Orthogonal Layouts

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 edges by circular arcs, such that adjacent segments have the same horizontal or vertical tangent at their contact points. 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 Fig. 3.

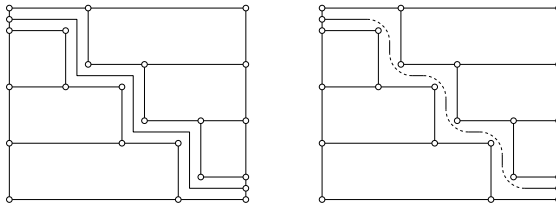


Fig. 3. Edge complexity might increase from k to $2k - 1$ for “staircase” edges

In practice, it might be possible to avoid increasing the edge complexity. One way to achieve this 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 vertex, assuming that all circular arcs have the same radius.

Lemma 1. *Given a fixed orthogonal layout, there exists an $O(N \log N)$ -algorithm that maximizes the radii of the circular arcs in the drawing, 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). Since the vertices have fixed positions, the corresponding decision problem can be answered in $O(N \log N)$ time by a plane-sweep method. This is enough to apply the randomized optimization technique of Chan to solve the problem in $O(N \log N)$ time [5]. \square

3 Smooth Postprocessing: 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. 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 [19]. Here, we call it the *fixed-shape model*. Even though this model is also restrictive, it provides us with enough flexibility to produce layouts with low edge complexity.

3.1 Layout Stretching

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. Here we use circular arcs of varying sizes (unlike in Lemma 1, where we used circular arcs of the same size).

Theorem 2. *Any complexity-2 orthogonal layout can be transformed into a smooth complexity-2 layout in linear time.*

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 . Then in the stretched drawing, the vertical segment is no larger than its matching horizontal segment. Also due to the stretching, we can easily verify that the grid of size $l \times l$ in each quadrant of vertex v is empty of other vertices and edges. Then we can safely replace the vertical segment with a quarter-circle arc, which yields a smooth complexity-2 realization of the edge.

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.

It is easy to verify that once this procedure has been used to modify all complexity-2 edges, the result is a smooth complexity-2 orthogonal layout that is a factor of n larger than the input layout. 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. \square

Corollary 3. *Any complexity-3 orthogonal layout can be transformed into a smooth complexity-4 orthogonal layout.*

Note that we can also apply the stretching technique to any edge with complexity $k > 1$; see Fig. 4a. For edges that turn only in the same direction (i.e., right, right, right) the edge complexity does not increase. This together with the fact that orthogonal layouts of 4-planar graphs with minimum number of bends do not have S-shaped edges [19], results into the following theorem:

Theorem 4. *Let G be a 4-planar graph that admits an orthogonal drawing H with minimum number of bends and edge complexity k . Then, G admits a smooth orthogonal layout H' of edge complexity k , whose total number of edge segments and arcs is not greater than the total number of edge segments of H .*

For edges that turn in alternating directions (i.e., staircase edges), the edge complexity increases from k to $3/2k$, since each vertical segment is replaced by an S-shape pattern with complexity 2.

Corollary 5. *Stretching increases edge complexity by no more than a 1.5 factor.*

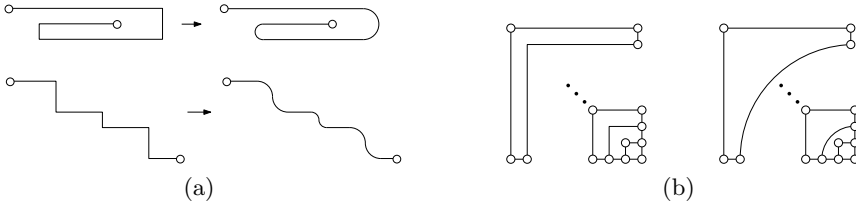


Fig. 4. (a) Illustration of horizontal stretching for edges with many bends: uniform turns do not increase complexity while alternating turns do. (b) Illustration of exponential area blow-up for complexity-1 drawings.

3.2 Area Bounds for Smooth Orthogonal Layouts

Traditional orthogonal layout algorithms place the vertices on an integer grid of size $O(n) \times O(n)$. When the stretching technique is applied the result is a smooth layout with increased layout area from $O(n^2)$ to $O(n^3)$. The situation is different if we want to generate smooth layouts with complexity exactly one. In particular, for smooth complexity-1 layout, the area penalty can be exponential.

Theorem 6. *There exists a graph whose smooth complexity-1 layout requires exponential area, in the fixed topology setting.*

Sketch of proof. The graph we use is shown in Fig. 4b. If every edge is to be drawn with complexity one, we must use quarter-circle arcs. Let a_i be the length of the i -th curve from the origin. It is not difficult to prove that $a_n = O(\sqrt{2^{\frac{n}{2}}})$. \square

4 Smooth Layouts with Low Complexity

While many 4-planar graphs have complexity-2 orthogonal layouts and can be transformed into smooth complexity-2 layouts, this is not true for all graphs. We show that all biconnected 4-planar graphs admit smooth complexity-3 layouts, which is next to optimal, as for smooth layouts complexity 2 is necessary.

Theorem 7. *Any biconnected 4-planar graph admits a smooth complexity-3 layout.*

Proof. Any 4-planar graph, except the octahedron, admits an orthogonal layout of edge complexity 3. This is due to an algorithm of Biedl and Kant [1]. In a high-level description, their algorithm for the case of 4-planar biconnected graphs, chooses a pair of vertices s and t of minimum degree and iteratively places the vertices of the input graph in the order implied by its st -numbering, maintaining the following invariant: At each iteration, every edge, for which exactly one endpoint is drawn, is associated with a column (vertical line). The first two vertices are drawn bottommost in the same horizontal row; see Fig. 5a. A vertex later in the order is drawn in the bottommost available row and occupies as many neighboring columns as necessary to maintain the invariant.

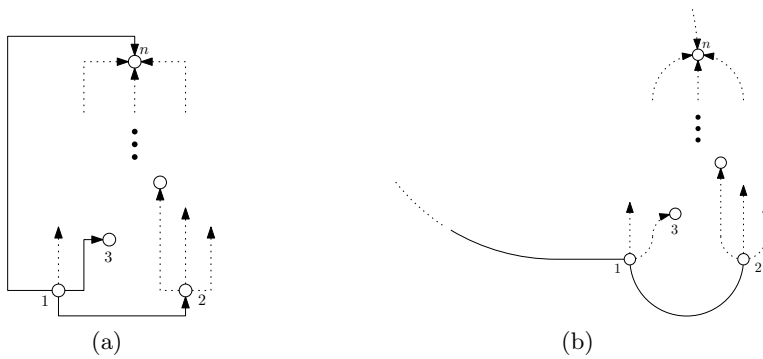


Fig. 5. Illustration of the Biedl-Kant algorithm [1] and its smooth modification

In the resulting layout, two edges are of particular importance: the bottom-most U-shape edge, and the left/right-most C-shape edge. The other edges are of edge complexity either one (drawn as straight-line vertical segments), or two (made of one vertical and one horizontal segments), or three (S-shapes, made of one vertical and two horizontal segments).

We modify the algorithm using the stretching technique above, so that every edge that is created has smooth complexity 3 or less. Here the stretching used is vertical, and ensures that we can replace every horizontal segment with a circular arc of appropriate radius, without introducing crossings; see Fig. 5b. For a complexity-2 edge, the stretch must ensure that the horizontal segment has smaller length than the vertical segment. For a complexity-3 edge, the stretch must ensure that the sum of the lengths of the two horizontal segments is less than the length of the vertical segment. This can be accomplished with a uniform vertical stretch of factor at most $2n$. The following cases based on the types of edges arise (all illustrated in the small example in Fig. 5).

A complexity-1 edge remains a straight-line segment after the uniform stretching. There are two types of complexity-2 edges. A complexity-2 edge, which initially is made of a straight-line horizontal segment followed by a vertical one, is transformed into an edge of smooth complexity two, consisting of a quarter-circle arc followed by a vertical straight-line segment. A complexity-2 edge, which initially is made of a straight-line vertical segment followed by a horizontal one, is transformed into an edge of smooth complexity two, consisting of a vertical straight-line segment followed by a quarter-circle arc.

All but two of the complexity-3 edges are S-shaped, initially made of a horizontal, vertical, horizontal segment sequence. This is transformed into an edge of smooth complexity three, consisting of a quarter-circle arc followed by a vertical straight-line segment and another quarter-circle arc; see Fig. 6. The bottom-most U-shape edge is transformed into an edge of smooth complexity 1 consisting of a half-circular arc. The left/right-most C-shape edge is transformed into an edge of smooth complexity 2 consisting of a horizontal straight-line segment followed by a 3/4-circular arc.

It is particularly important that we never encounter a “staircase” edge of the type that increases the complexity when its layout is transformed from orthogonal to smooth. The absence of such edges can be guaranteed by the layout construction of the Biedl-Kant algorithm [1]. In order to complete the proof of this Theorem, it remains to prove that the octahedron graph admits a complexity-3 layout as well, which is trivial (see Figure 1g). Note that the algorithm of Biedl and Kant can be adjusted to support non-biconnected graphs as well. To do so, the graph is split into blocks which are embedded separately and appropriately merged. Unfortunately, we were not able to support this case, since the blocks are rotated, flipped etc. \square

5 Smooth Complexity-1 Layouts

The use of circular arcs allows us to not only 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 Fig. 1. However, we cannot always achieve this, as shown in the next theorem.

Theorem 8. *There exists a 4-planar graph that does not admit a smooth layout with complexity one.*

Proof. 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 outer-face. The outer-face 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 outer-face can be a straight-line segment or a quarter-circle arc (see Fig. 7a). In fact, the only way to realize the face and keep the ports inside is with two

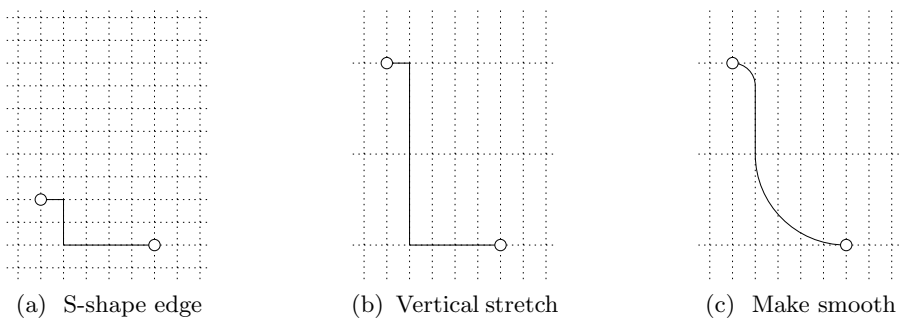


Fig. 6. Modifying the layout of a complexity-3 edge

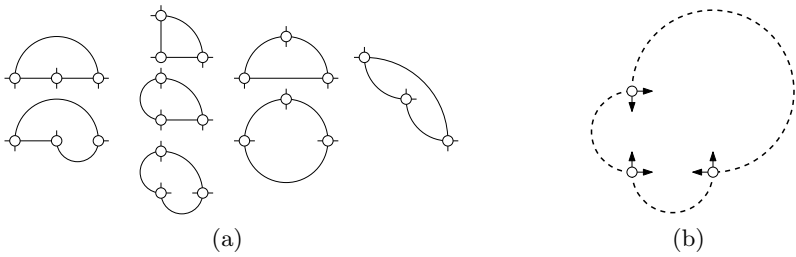


Fig. 7. (a) Different realizations of the triangle graph with edge complexity one, in which quarter-circle arcs can be appropriately replaced by 3/4-circle arcs to obtain all different realizations of the triangle graph. (b) A realization with two half-circle arcs and one 3/4-circle arc in which the ports are kept inside.

half-circle arcs and one 3/4-circle arc (see Fig. 7b). Moreover, this feasible configuration is unique, up to rotation and scaling.

Note that the only feasible realization of the outer-face 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 outer-face and leave two free ports pointing inside. Using careful case analysis, it is not difficult to show that there is no feasible place for such a vertex.

In traditional orthogonal setting, the octahedron is the only graph that requires complexity 4. That is, all other 4-planar graphs can be drawn with complexity 3 or less. In the smooth setting we can draw the octahedron with complexity 2, while the above theorem shows that there is no smooth complexity-1 layout. While tempting to think otherwise, it is significant to note here that the class of 4-regular planar graphs that do not admit smooth complexity-1 layouts is infinite, as the next corollary shows.

Corollary 9. *There exist infinitely many 4-planar graphs that do not admit smooth complexity-1 layouts.*

Proof. We construct a class of 4-planar graphs with a 4-regular triangular outer-face 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 outer-face. 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 outer-face. Each of these special vertices must use consecutive ports to connect to the outer-face 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 outer-face under these constraints, and use only complexity-1 edges.

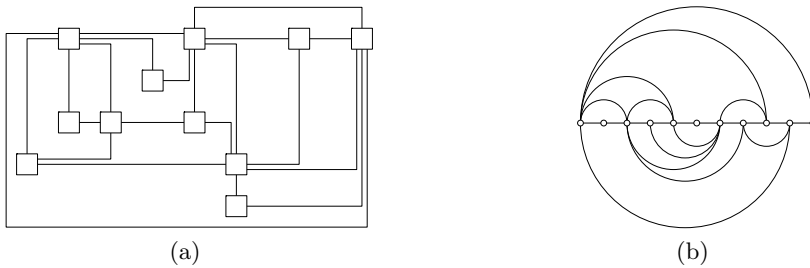


Fig. 8. (a) A *podevsef* style layout [10]; (b) A smooth Kandinsky layout of the same graph

Corollary 9 suggests that not all 4-planar graphs admit complexity one smooth orthogonal layouts. So, we turn our attention in the case of 3-planar graphs, for which we identify a class that admits complexity one smooth orthogonal layouts.

Theorem 10. *Any triconnected 3-planar graph admits a smooth complexity-1 layout.*

Sketch of proof. Due to space constraints, the detailed proof is omitted. The algorithm initially constructs an orthogonal layout (by appropriately modifying an algorithm of Kant [15]), which in turn, is transformed into a smooth orthogonal layout in the second phase of the algorithm. \square

6 Smooth Orthogonal Layouts for High Degree Graphs

A serious limitation for the practical applicability of orthogonal layouts and consequently for smooth orthogonal layouts, is the vertex degree restriction. Several extensions that overcome this restriction have been proposed for orthogonal layouts [19]. A quite common approach is the *podefsnev* model [10], also known as the Kandinsky model [2]. 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. Such edges conceptually end at one point, and are drawn ending next to each other; see Fig. 8a. The edges follow the *bend-or-end condition*, which means that for edge e attached at the top of v and a node w placed directly above v at the same vertical grid-line, either edge e bends to the left or right before intersecting w or it ends at the bottom side of w .

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. As a layout algorithm suitable for this model, we use the technique of Kaufmann and Wiese [16] for point-set embedding with few bends per edge. If the graph is Hamiltonian, then we can distribute the vertices evenly along a horizontal line, in the order given by the Hamiltonian path. Vertices adjacent on the path are connected with edges that are straight-line segments. All other edges are attached either at the top or at the bottom side

of the vertices and are connected by half-circles. The Hamiltonian cycle splits the remaining edges of the graph into two groups: one with edge inside the cycle and the other with edges outside the cycle. We route these edges using half-circle arcs above and below the horizontal line containing the vertices, respectively. This can be accomplished using edges of complexity one; see Fig. 8b.

For non-Hamiltonian graphs, we use the technique of Kaufmann and Wiese of splitting separating triangles by inserting dummy vertices on appropriate edges and triangulating 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 it might consist of two segments, an upper and a lower half-circle. This results in a smooth layout with edges of complexity at most two, which yields the theorem below.

Theorem 11. *Any planar graph admits a smooth complexity-2 layout in the Kandinsky model.*

Corollary 12. *Any planar Hamiltonian graph admits a smooth complexity-1 layout in the Kandinsky model.*

Corollary 13. *Any planar Hamiltonian degree-3 graph can be drawn with complexity one in the smooth orthogonal model.*

7 Conclusion and Open Problems

In this paper, we introduced and presented the first combinatorial results for smooth orthogonal layouts in the context of improved drawing readability and drawing aesthetics. 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. We proved that any biconnected 4-planar graph admits a smooth complexity-3 layout. However, it remains open whether the same holds for the case of non-biconnected graphs. 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 $k > 1$?

Acknowledgments. We would like to thank M. Nöllenburg and I. Rutter for helpful discussions.

References

1. Biedl, T., Kant, G.: A Better Heuristic for Orthogonal Graph Drawings. In: van Leeuwen, J. (ed.) ESA 1994. LNCS, vol. 855, pp. 24–35. Springer, Heidelberg (1994)
2. Biedl, T.C., Kaufmann, M.: Area-Efficient Static and Incremental Graph Drawings. In: Burkard, R.E., Woeginger, G.J. (eds.) ESA 1997. LNCS, vol. 1284, pp. 37–52. Springer, Heidelberg (1997)

3. Brandes, U., Schlieper, B.: Angle and Distance Constraints on Tree Drawings. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 54–65. Springer, Heidelberg (2007)
4. Brandes, U., Wagner, D.: Using graph layout to visualize train interconnection data. *J. of Graph Algorithms and Applications* 4(3), 135–155 (2000)
5. Chan, T.M.: Geometric applications of a randomized optimization technique. *Discrete Computational Geometry* 22(4), 547–567 (1999)
6. Cheng, C.C., Duncan, C.A., Goodrich, M.T., Kobourov, S.G.: Drawing planar graphs with circular arcs. *Discrete Comput. Geom.* 25(3), 405–418 (2001)
7. Duncan, C.A., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Löffler, M.: Planar and Poly-arc Lombardi Drawings. In: van Kreveld, M., Speckmann, B. (eds.) GD 2011. LNCS, vol. 7034, pp. 308–319. Springer, Heidelberg (2012)
8. Duncan, C.A., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Nöllenburg, M.: Lombardi Drawings of Graphs. *J. of Graph Algorithms and Applications* 16(1), 85–108 (2011)
9. Finkel, B., Tamassia, R.: Curvilinear Graph Drawing Using the Force-Directed Method. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 448–453. Springer, Heidelberg (2005)
10. Fößmeier, U., Kaufmann, M.: Drawing High Degree Graphs with Low Bend Numbers. In: Brandenburg, F.J. (ed.) GD 1995. LNCS, vol. 1027, pp. 254–266. Springer, Heidelberg (1996)
11. Garg, A., Tamassia, R.: On the Computational Complexity of Upward and Rectilinear Planarity Testing. In: Tamassia, R., Tollis, I.G. (eds.) GD 1994. LNCS, vol. 894, pp. 286–297. Springer, Heidelberg (1995)
12. Goodrich, M.T., Wagner, C.G.: A framework for drawing planar graphs with curves and polylines. *J. of Algorithms* 37(2), 399–421 (2000)
13. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. *Computer Graphics Forum* 28(3), 983–990 (2009)
14. Huang, W., Eades, P., Hong, S.H.: A graph reading behavior: Geodesic-path tendency. In: IEEE Pacific Visualization Symp. (PacificVis 2009), pp. 137–144. IEEE (2009)
15. Kant, G.: Drawing planar graphs using the canonical ordering. *Algorithmica* 16(1), 4–32 (1996)
16. Kaufmann, M., Wiese, R.: Embedding vertices at points: Few bends suffice for planar graphs. *J. of Graph Algorithms Applications* 6(1), 115–129 (2002)
17. Leiserson, C.E.: Area-efficient graph layouts (for VLSI). In: 21st Annual IEEE Symp. on Foundations of Computer Science, vol. 1547, pp. 270–281. IEEE (1980)
18. Lombardi, M., Hobbs, R.: Mark Lombardi: Global Networks. Independent Curators (2003)
19. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM J. of Computing* 16(3), 421–444 (1987)
20. Tamassia, R., Tollis, I.: Planar grid embedding in linear time. *IEEE Transactions on Circuits and Systems* 36(9), 1230–1234 (1989)
21. Valiant, L.G.: Universality considerations in VLSI circuits. *IEEE Transaction on Computers* 30(2), 135–140 (1981)