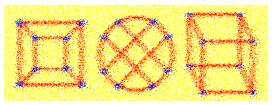
Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 15, no. 1, pp. 79–96 (2011)

Geometric Simultaneous Embeddings of a Graph and a Matching

Sergio Cabello[1] *Marc van Kreveld*[2] *Giuseppe Liotta*[3]
Henk Meijer[4] *Bettina Speckmann*[5] *Kevin Verbeek*[5]

[1]Faculty of Mathematics and Physics, University of Ljubljana, Slovenia
[2]Department of Information and Computing Sciences,
Utrecht University, The Netherlands
[3]Dipartimento di Ingegneria Elettronica e dell'Informazione,
Università di Perugia, Italy
[4]Roosevelt Academy, Middelburg, The Netherlands
[5]Department of Mathematics and Computer Science,
TU Eindhoven, The Netherlands

Abstract

The geometric simultaneous embedding problem asks whether two planar graphs on the same set of vertices in the plane can be drawn using straight lines, such that each graph is plane. Geometric simultaneous embedding is a current topic in graph drawing and positive and negative results are known for various classes of graphs. So far only connected graphs have been considered. In this paper we present the first results for the setting where one of the graphs is a matching.

In particular, we show that there exist a planar graph and a matching which do not admit a geometric simultaneous embedding. This strengthens an analogous negative result for a planar graph and a path. On the positive side, we describe algorithms that compute a geometric simultaneous embedding of a matching and a wheel, outerpath, or tree. Our drawing algorithms minimize the number of orientations used to draw the edges of the matching. Specifically, when embedding a matching and a tree, we can draw all matching edges horizontally. When embedding a matching and a wheel or an outerpath, we use only two orientations.

Submitted:	Reviewed:	Revised:	Accepted:
November 2009	August 2010	September 2010	October 2010
	Final:	Published:	
	November 2010	February 2011	
Article type:		Communicated by:	
Regular paper		D. Eppstein and E. R. Gansner	

E-mail addresses: sergio.cabello@fmf.uni-lj.si (Sergio Cabello) marc@cs.uu.nl (Marc van Kreveld) liotta@diei.unipg.it (Giuseppe Liotta) h.meijer@roac.nl (Henk Meijer) speckman@win.tue.nl (Bettina Speckmann) k.a.b.verbeek@tue.nl (Kevin Verbeek)

1 Introduction

The computation of node-link diagrams of two sets of relations on the same set of data is a recent and already well-established research direction in network visualization. The interest in this problem is partly due to its theoretical relevance and partly motivated by its importance in many application areas, such as software engineering, databases, and social networks. There are various application scenarios where a visual analysis of dynamic and evolving graphs defined on the same set of vertices is useful, see [5, 6] for detailed descriptions.

Formally, the problem can be stated as follows: Let G_1 and G_2 be two graphs that share their vertex set, but which have different sets of edges. We would like to compute two readable drawings of G_1 and G_2 such that the locations of the vertices are the same in both visualizations. Cognitive experiments [11] prove that the readability of a drawing is negatively affected by the number of edge crossings and by the number of bends along the edges. Hence, if G_1 and G_2 are both planar, we want to compute plane drawings of the two graphs where the vertices have the same locations and edges are straight-line segments. Note that we allow edges from different graphs to cross.

In a seminal paper, Brass *et al.* define a *geometric simultaneous embedding* of two planar graphs sharing their vertex set as two crossing-free straight-line drawings that share the locations of their vertices [2]. Geometric simultaneous embedding is a current topic in graph drawing and positive and negative results are known for various classes of graphs. A comprehensive list can be found in Table 1 of a recent paper by Frati, Kaufmann, and Kobourov [9]. Specifically, Brass *et al.* [2] show that two paths, two cycles, and two caterpillars always admit a geometric simultaneous embedding. (A caterpillar is a tree such that the graph obtained by deleting its leaves is a path.) The authors also prove that three paths may not admit a geometric simultaneous embedding. Erten and Kobourov [7] prove that a planar graph and a path may not admit a geometric simultaneous embedding. Frati, Kaufmann, and Kobourov [9] extend this negative result to the case where the path and the planar graph do not share any edges. Geyer, Kaufmann, and Vrt'o [10] show that two trees may not have a geometric simultaneous embedding. A major open question in this area was solved very recently: Angelini *et al.* [1] show that a tree and a path may not have a geometric simultaneous embedding. Finally, Estrella-Balderrama *et al.* [8] prove that determining whether two planar graphs admit a geometric simultaneous embedding is NP-hard.

So far, only connected graphs have been considered and in particular, there are no results for one of the simplest classes of graphs, namely *matchings*. A matching is an independent set of edges. Clearly a geometric simultaneous embedding of two matchings always exists, since the union of two matchings is a collection of cycles and hence planar. But already the union of the edges of a path and a matching is not always planar: Fig. 1 (left) shows a path and a matching which form a subdivision of $K_{3,3}$.

Results. We study geometric simultaneous embeddings of a matching with various standard classes of graphs. In Section 2 we show that there exists a planar graph and a matching which do not admit a geometric simultaneous embedding. This strengthens an analogous negative result for a planar graph and a path [7].

On the positive side, we describe algorithms that compute a geometric simultaneous embedding of a matching and a wheel, outerpath, or tree. Specifically, in Section 3 we sketch a construction that computes a geometric simultaneous embedding of a wheel and a cycle, which immediately implies an embedding for a wheel and a matching. In Section 4 and 5 we describe algorithms to embed a matching together with two specific types of outerplanar graphs, namely *outerzigzags* and *outerpaths*. An outerzigzag is also known as a triangle strip. Its weak dual is a path and each of its vertices has degree at most 4. An outerpath is an outerplanar graph whose weak dual is a path. Our result for outerpaths subsumes the result for outerzigzags, but we nevertheless first present the construction for outerzigzags, to introduce our techniques on a conceptually simpler class of graphs. The algorithms for the wheel, the outerzigzag, and the outerpath, preserve the "natural" embedding of these graphs. That is, the center of the wheel is not incident to the outer face, and the embeddings of the outerplanar graphs are outerplanar. Note that an outerplanar graph and a path may not have a geometric simultaneous embedding if the circular ordering of the edges around the vertices of the outerplanar graph is fixed a-priori [9]. In Section 6 we present an algorithm that computes a geometric simultaneous embedding of a tree and a matching. This algorithm is inspired by and closely related to an algorithm by Di Giacomo *et al.* [3]. Finally, in Section 7 we briefly consider the simultaneous embeddability of three or more matchings.

All our drawing algorithms minimize the number of orientations used to draw the edges of the matching. This may simplify the visual inspection of the data and of their relationships in practice. Consider the simple example in Fig. 1 (right). It immediately shows that a geometric simultaneous embedding of an outerpath or wheel with a matching requires the matching edges to have at least two orientations. Our constructions match this bound. When embedding a matching and a tree, we can even draw all matching edges horizontally.

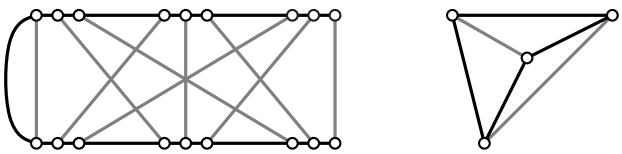

Figure 1: Left: The union of a path (black) and a matching (gray) can be non-planar. Right: Two orientations of the matching edges (gray) are forced.

2 Planar graph and matching

Theorem 1 *There exists a planar graph and a matching that do not admit a geometric simultaneous embedding.*

Proof: We denote the planar graph by G and the matching by M, refer to Fig. 2 for further notation. Consider the subgraph G_1 of G and the submatching M_1 of M which are induced by the vertices a, a', b, b', c, and c', marked by boxes in the figure. Correspondingly, let G_2 be the subgraph and let M_2 be the submatching induced by the remaining vertices. Note that the embedding subproblems $\langle G_1, M_1\rangle$ and $\langle G_2, M_2\rangle$ are isomorphic.

We first argue that there is no geometric simultaneous embedding of G_1 and M_1 where a, b, and c define the outer face of G_1. Indeed, if a, b, and c define the outer face of G_1, then all the other vertices of G_1 lie inside the triangle $\triangle abc$. Therefore, the matching edge (a, a') is also contained in $\triangle abc$, and thus some part of (a, a') is contained either in $\triangle abc'$ or $\triangle ac'c$. In the latter case, we would have a crossing between the edges (c, c') and (a, a'), and therefore we conclude that (a, a') must cross (b, c'). By symmetry, the edge (b, b') must cross (a, c'). However, this implies that (a, a') and (b, b') cross inside $\triangle abc'$.

Again by symmetry, there is no geometric simultaneous embedding of G_2 and M_2 where x, y, and z define the outer face of G_2. However, since G is 3-connected, any planar embedding of G has the same facial cycles as the embedding depicted in Fig. 2. This implies that in any planar embedding of G, either G_1 has $\triangle abc$ as the outer face, or G_2 has $\triangle xyz$ as the outer face, or both. In the first case, we have a crossing between two edges of M_1, in the second case, between two edges of M_2. □

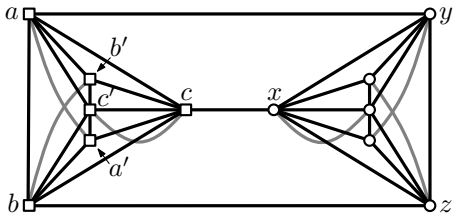

Figure 2: A planar graph (black) and a matching (gray) that do not admit a geometric simultaneous embedding.

3 Wheel and matching

We describe how to compute a geometric simultaneous embedding of a wheel and a cycle, which immediately implies the result for a wheel and a matching.

Theorem 2 *A wheel and a cycle always admit a geometric simultaneous embedding.*

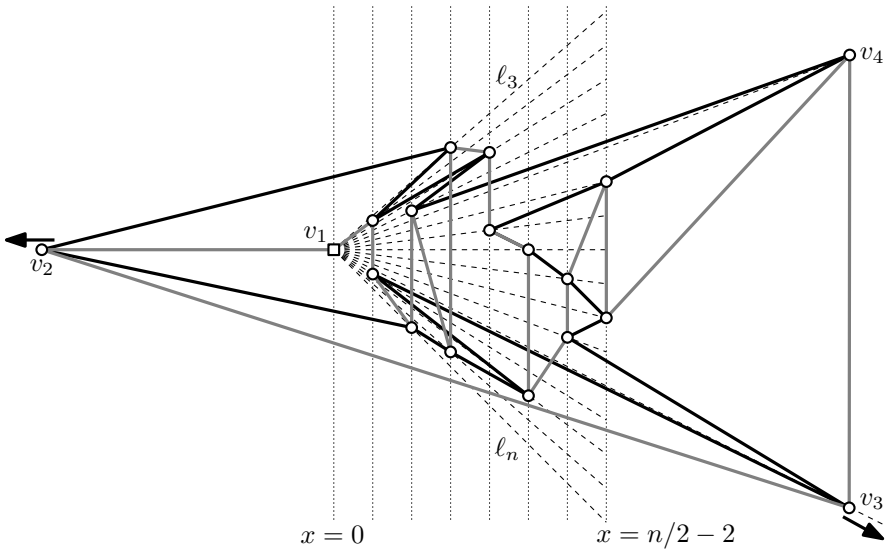

Figure 3: A geometric simultaneous embedding of a wheel and a cycle.

Proof: We denote the wheel by W and the cycle by C. They share a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$. Let v_1 be the center of W and let $v_1 \to v_2 \to \cdots \to v_n \to v_1$ be the edges of C. See Fig. 3: the center v_1 of the wheel is marked with a box, the rim is drawn in black, and the spokes are not drawn at all, although they do lie on the dashed lines. The cycle is drawn in gray.

We place v_1 at the origin and v_2 on the line $y = 0$ to the left of v_1. Next we place the remaining vertices on a set of rightward directed rays ℓ_i $(3 \leq i \leq n)$ which emanate from the origin. We assume that the rays are ordered clockwise, see Fig. 3: the dashed lines indicate the rays. The vertices are placed on the rays, according to their order along the rim of W. Their x-coordinates are determined by their order in C, as described below. Let v_3 be the other vertex connected to v_2 in C. We assign x-coordinates to the vertices such that the path obtained by removing (v_2, v_3) from C is drawn as an x-monotone curve. We use every x-coordinate twice, so that about half the edges of the cycle are drawn vertically. Finally, to remove any possible crossings with edge (v_2, v_3) of C, we move v_2 to the left and v_3 outwards along its ray (see Fig. 3). If the edge (v_3, v_4) was vertical and we would like to keep that property, then we need to do the same for v_4, which is always possible if we choose the rays for v_3 and v_4 on opposite sides of the x-axis. □

As stated above this construction immediately implies the result for a wheel and a matching: we can simply add edges to the matching to turn it into a cycle. When drawing a matching in this fashion, all matching edges are drawn vertically, with the exception of the first edge (v_1, v_2) which is drawn horizontally. Note that this edge is necessarily shared with a spoke of the

wheel. So we use exactly two orientations for the edges of the matching, which is optimal in the worst case.

4 Outerzigzag and matching

Recall that an outerzigzag is a triangle strip: it is a triangulated outerplanar graph, whose weak dual is a path and whose vertices have degree at most 4. More precisely, there are exactly two vertices of degree 2, two vertices of degree 3, and all other vertices have degree 4. Let $G_1 = (V, E_1)$ be an outerzigzag and let $G_2 = (V, E_2)$ be a matching. We first place the vertices of V in such a way that their placement induces a plane drawing of G_1. We then move some of the vertices vertically to planarize G_2, while keeping the drawing of G_1 planar.

Specifically, we initially place the vertices of V at positions $(0, 0)$, $(2, 1)$, $(4, 0)$, $(6, 1)$, $(8, 0)$, etc., and adjust their vertical positions as needed such that they remain on a grid of size $2n \times 4n$. One of the degree-2 vertices of G_1 is drawn at $(0, 0)$, the remainder is drawn in such a way that the edges of G_1 always connect two consecutive vertices on the line $y = 0$, or two consecutive vertices on the line $y = 1$, or two vertices at distance $\sqrt{5}$, see Fig. 4.

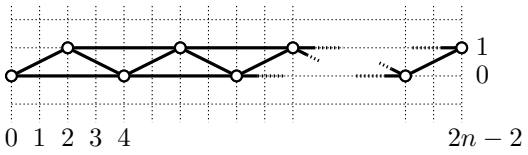

Figure 4: Drawing an outerzigzag G_1 on a grid.

We classify the edges of the matching G_2 based on the placement of their vertices on the grid:

BB-edges connect any two vertices on the line $y = 0$.

TT-edges connect any two vertices on the line $y = 1$.

BT-edges connect two vertices $(i, 0)$ and $(j, 1)$ with $i < j$.

TB-edges connect two vertices $(i, 1)$ and $(j, 0)$ with $i < j$.

We then move half of the vertices of V vertically according to three simple rules:

1. Only the right vertex of a matching edge moves, the left vertex is fixed.
2. The right vertex of every BT- and TT-edge is moved up until the edge has slope $+1$.
3. The right vertex of every TB- and BB-edge is moved down until the edge has slope -1.

See Fig. 5 for an example. It is easy to see that the displacements preserve the planarity of the embedding of G_1, because vertices only move vertically. The displacements also make G_2 planar: all edges of E_2 with slope $+1$ are on parallel diagonal lines, so they cannot intersect. Symmetrically, all edges of E_2 with slope -1 cannot intersect. Finally, an edge with slope -1 and an edge with slope $+1$ from E_2 cannot intersect because their only y-overlap is between 0 and 1, and here they are sufficiently separated to prevent intersections. Clearly this construction uses only two orientations for the edges of the matching.

Theorem 3 *An outerzigzag and a matching always admit a geometric simultaneous embedding.*

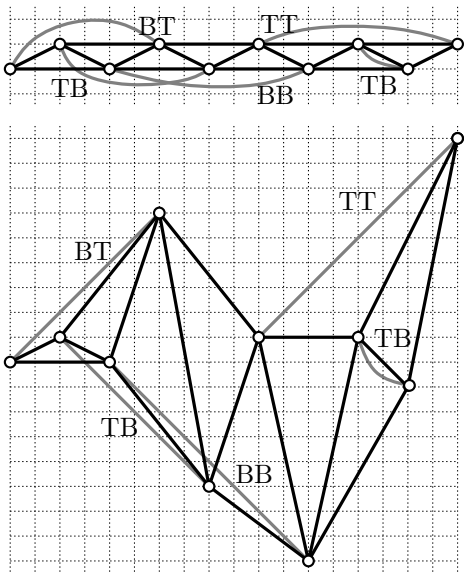

Figure 5: A geometric simultaneous embedding of an outerzigzag (black) and a matching (gray).

5 Outerpath and matching

We now extend the approach for outerzigzags to outerpaths. First, we assume that the outerpath is triangulated. Since a triangulated outerpath has two vertices of degree 2, we can place them as the leftmost and rightmost vertex of an initial placement. We place all vertices on two horizontal lines $y = 0$ and $y = 1$ in such a way that we obtain a plane drawing of the outerpath G_1 (see Fig. 6). We say that vertices which lie on the line $y = 1$ are on the *top chain*, correspondingly, vertices which lie on the line $y = 0$ are on the *bottom chain*. The leftmost vertex is placed at $(0, 0)$. In the final drawing the edges of the matching have slopes -1 or $+1$ as before. However, the embedding algorithm for outerpaths needs to move vertices not only vertically, but also horizontally and hence the x-order of the vertices in the initial placement is not preserved.

We view an outerpath as a sequence of *maximal fans* that are alternatingly directed upwards and downwards. A maximal fan shares its first and last edge

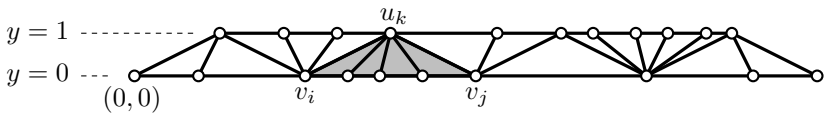

Figure 6: Outerpath with one fan indicated in gray.

with a neighboring fan. Fig. 6 shows a downward fan in gray. We denote by d the maximum degree of any vertex in the outerpath G_1.

Our algorithm works as follows: we treat one fan after the other, moving from left to right. When we treat a fan, we place its vertices at new locations to planarize G_2, while keeping the drawing of G_1 planar. In the following we explain the placement algorithm for a downward fan F, upward fans are treated similarly. We denote the single *apex* vertex of F by u_k and its sequence of *finger* vertices by $v_i, \ldots, v_j$ (see Fig. 6). Note that $i < j$; if $i = j$ then the outerpath was not triangulated or F was not maximal. We place all vertices of F, with the exception of v_i and v_j. Vertex v_i has already been placed, since it is the apex of the preceding fan (or it is the leftmost vertex which remains fixed). We do not place v_j since it is the apex of the following fan and will be placed when that fan is treated. We distinguish three cases, depending on the matching partner of the apex u_k. Case (1): the matching partner of u_k has already been placed, Case (2): the matching partner of u_k has not been placed yet and it is not among $v_{i+1}, \ldots, v_{j-1}$, and Case (3): the matching partner of u_k is among $v_{i+1}, \ldots, v_{j-1}$.

Case (1) Apex u_k has a matching partner that has already been placed. Hence the matching partner lies either on the top chain and has an index smaller than k, or it lies on the bottom chain and has an index smaller than or equal to i. Let X denote the total width (x-range) of the drawing constructed so far. We place u_k at x-coordinate $2X + 1$ and then move u_k upwards until it lies on the line with slope $+1$ through its matching partner (see Fig. 7 (left)).

Next we place $v_{i+1}, \ldots, v_{j-1}$ at positions $(2X, 0)$, $(2X + 1/d, 0)$, …, $(2X + (j-i-2)/d, 0)$. Consider the $j-i-1$ lines through u_k and each of $v_{i+1}, \ldots, v_{j-1}$. If we ensure that the final placements of $v_{i+1}, \ldots, v_{j-1}$ lie on these lines, then

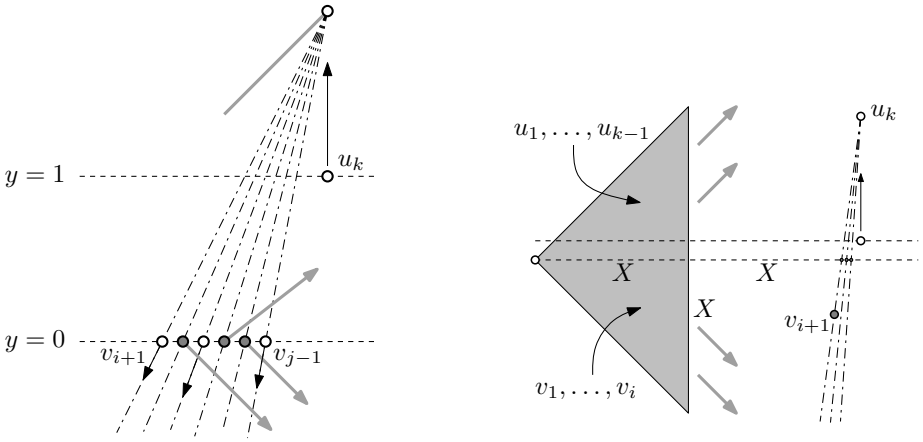

Figure 7: Left: Case (1), u_k is a right vertex of a matching edge. Right: Global situation of Case (1), previously placed vertices lie inside the gray triangle.

we will never invert any triangle of the fan and hence keep the drawing of G_1 planar. We now move those vertices of $v_{i+1}, \ldots, v_{j-1}$ that are right vertices of matching edges down on their lines until they reach the proper position, determined by the lines with slope -1 through their matching partners. Those vertices of $v_{i+1}, \ldots, v_{j-1}$ that are left vertices of matching edges stay where they are; they define lines with slope -1 or $+1$ on which their matching partners will be placed eventually. By construction, none of these lines intersect each other to the right of the vertices that defined them, and they also do not intersect the lines defined by vertices treated earlier (see Fig. 7 (left)).

See Fig. 7 (right) for a global sketch of Case (1). Note that v_{i+1} stays to the right of all vertices placed before. This is true because the line defined by u_k and v_{i+1} has slope > 1, and the separation between v_{i+1} and the previously placed vertices is at least X. The y-range for the previously placed vertices is $[-X, +X]$, since the edges of the matching have slopes -1 and $+1$. Further note that triangle $\triangle u_k v_{i+1} v_i$ is not inverted, regardless of where v_i is placed in the initial part and whether v_{i+1} is moved on its line. Finally, note that v_j can be placed anywhere on the line $y = 0$ or lower, as long as its x-coordinate is at least that of u_k: the triangle $\triangle u_k v_j v_{j-1}$ will not be inverted.

Case (2) Apex u_k has a matching partner that has not been placed yet and which is not among $v_{i+1}, \ldots, v_{j-1}$. We place u_k at position $(3X + 2, 1)$, where X is again defined as the total width of the drawing constructed so far. Next we place the vertices $v_{i+1}, \ldots, v_{j-1}$ at positions $(3X, 0)$, $(3X + 1/d, 0)$, …, $(3X + (j-i-2)/d, 0)$. Consider the $j-i-1$ lines through u_k and each of $v_{i+1}, \ldots, v_{j-1}$. We again move those vertices of $v_{i+1}, \ldots, v_{j-1}$ that are right vertices of matching edges down on their lines until they reach the proper position, determined by the lines with slope -1 through their matching partners (see Fig. 8). All lines on which the vertices move have slope at least $1/2$, implying that all vertices of $v_{i+1}, \ldots, v_{j-1}$ are placed to the right of all previously placed vertices, due to the x-separation of at least $2X$. Again we note that $\triangle u_k v_{i+1} v_i$ is not inverted, and that v_j may be placed anywhere to the right of u_k without the risk of inverting $\triangle u_k v_j v_{j-1}$.

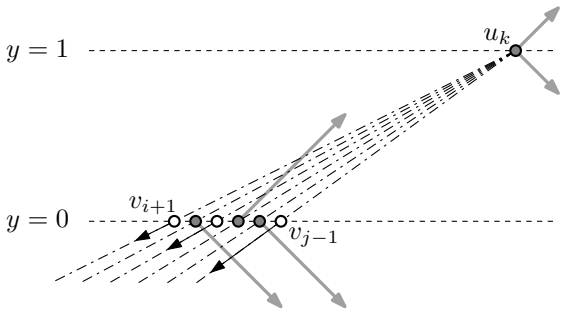

Figure 8: Case (2), u_k is a left vertex of a matching edge.

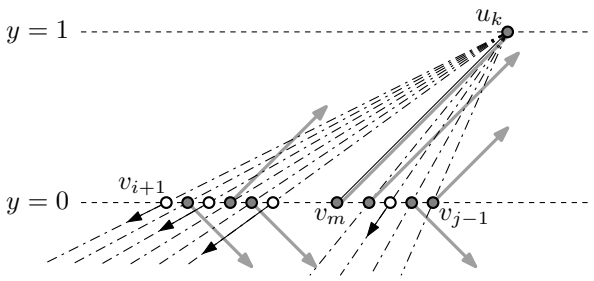

Figure 9: Case 3.

Case (3) Apex u_k has a matching partner v_m that is among $v_{i+1}, \ldots, v_{j-1}$, see Fig. 9. We place u_k at position $(3X + 2, 1)$ and v_m at position $(3X + 1, 0)$, where X is again the total width of the drawing constructed so far. Note that the edge (u_k, v_m) is an edge of both G_1 and G_2. Next we place the vertices $v_{i+1}, \ldots, v_{m-1}$ at positions $(3X, 0)$, $(3X + 1/d, 0)$, ..., $(3X + (m - i - 2)/d, 0)$, and the vertices $v_{m+1}, \ldots, v_{j-1}$ at positions $(3X+1+1/d, 0)$, $(3X+1+2/d, 0)$, ..., $(3X+1+(j-m-1)/d, 0)$. As before, we now use the lines through u_k and each of $v_{i+1}, \ldots, v_{m-1}, v_{m+1}, \ldots, v_{j-1}$ to move vertices down if they are right vertices of matching edges.

Theorem 4 *An outerpath and a matching always admit a geometric simultaneous embedding.*

6 Tree and matching

Our algorithm that computes a geometric simultaneous embedding for a tree and a matching is inspired by and closely related to an algorithm by Di Giacomo *et al.* [3], which computes a *matched drawing* of two trees. Matched drawings are a relaxation of geometric simultaneous embeddings. Specifically, two planar graphs G_1 and G_2 are *matched*, if they are defined on two vertex sets V_1 and V_2 of the same cardinality and there is a one-to-one mapping between V_1 and V_2. A matched drawing of two matched graphs is a pair of planar straight-line drawings, such that matched vertices of G_1 and G_2 are assigned the same y-coordinate. A geometric simultaneous embedding of a tree and a matching is in essence a matched drawing of half of the vertices of the tree with the other half. And indeed, the algorithm by Di Giacomo *et al.* can be adapted to compute a geometric simultaneous embedding of a tree and a matching. However, the edges of the matching in the resulting drawing will in general not all have the same orientation. In the remainder of this section we give a construction akin to the one from [3], to compute a simultaneous embedding where all matching edges are drawn horizontally.

Definitions and overview. We place the vertices one by one, always placing the two vertices of a matching edge consecutively at the same y-coordinate. We

use y-coordinates $1, \ldots, n/2$, from the outside in. That is, at any point of the construction, there are two indices i and j with $1 \leq i \leq j \leq n/2$ such that the coordinates $1, \ldots, i-1$ and $j+1, \ldots, n/2$ have been used, and the coordinates $i, \ldots, j$ have not been used yet. Before every odd-numbered placement we decide if we should place the next vertex at the top or at the bottom, that is, at the highest or the lowest available y-coordinate. The next, even-numbered placement will be at the same y-coordinate.

Let T be the tree with some of its vertices already placed. The placed vertices induce an edge partition of the tree into connected components (subtrees); we call each component up to and including the placed vertices a *rope*. The placed vertices incident to a rope are called the *knots* of that rope, see Fig. 10. Hence, placed vertices will be knots of several ropes, exactly as many as the degree of the vertex in the tree. We maintain the following invariant: after every odd placement, every rope of T has one or two knots, but not more. After an even placement this invariant might be false for exactly one rope, which has three knots. Then there is a unique vertex, which we call the *splitter*, that lies on the three paths between the knots. We show below how to restore the invariant with the next odd placement by choosing the splitter as the next vertex to place. In the example of Fig. 10, vertex s became a splitter when v_6 was placed, so s will be placed next (and afterwards its matching partner). We define the *degree of a rope* to be the number of knots it has.

Since we place vertices from the outside in, and ropes have degree at most three, there are nine types of ropes which we can encounter during the construction. They are the degree-1 ropes with one knot at the top or at the bottom, the degree-2 ropes with two knots at the top, or two at the bottom, or one at the top and one at the bottom, and the degree-3 ropes with zero, one, two, or three knots at the top and three, two, one, or zero knots at the bottom, respectively. We call these ropes T-rope, B-rope, TT-rope, BB-rope, TB-rope, BBB-rope,

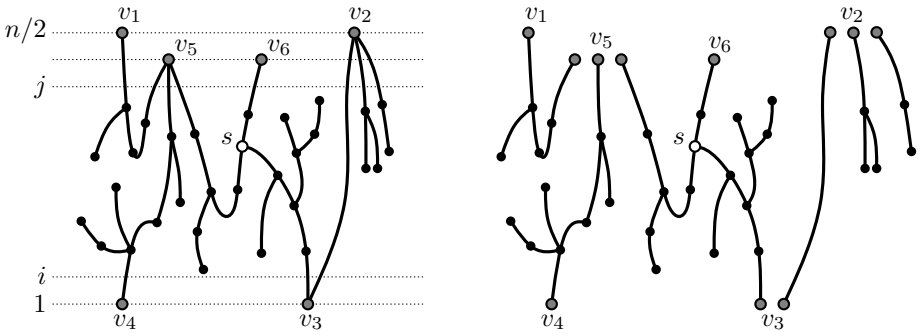

Figure 10: Left: A tree of which six vertices have been placed, where (v_1, v_2), (v_3, v_4), and (v_5, v_6) are edges of the matching. Right: The ropes that are induced. From left to right a TT-rope (with knots v_1 and v_5), a TB-rope (with knots v_5 and v_4), a TTB-rope, a TB-rope, and two T-ropes.

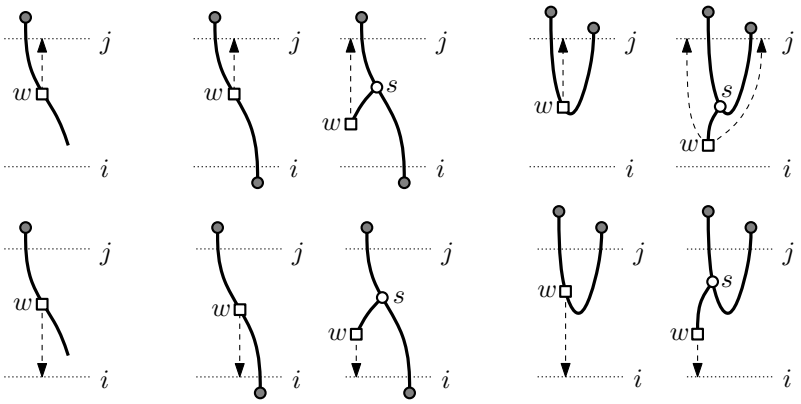

Figure 11: Even placement for T-ropes, TB-ropes, and TT-ropes.

TBB-rope, TTB-rope, or TTT-rope, respectively. A placement typically splits a rope into several ropes.

The constructive proof describes the placements in two steps. In the first step, we decide which vertex to place next, how to order the incident edges of this vertex, and what the left-to-right sequence of ropes becomes. We distinguish even and odd placements. In the second step, we show that we can associate suitable regions of the plane in which the ropes can be drawn, so that the tree is drawn plane and straight-line. We maintain these regions at each placement. In a sense, the first step assigns a y-coordinate, and the second step shows that an x-coordinate can be assigned as well.

Even placement. The invariant given above implies that before an even placement there are only degree-1 and degree-2 ropes. Furthermore, there is exactly one edge (v, w) of the matching M that has one, but not both of its vertices placed. We assume that v has been placed and we place w next, at the same y-coordinate as v. The exact placement depends on the type of rope w is part of, as well as the y-coordinate of v. Fig. 11 shows the cases for T-ropes, TB-ropes, and TT-ropes; B-ropes and BB-ropes are symmetric. Placing w can create at most two degree-2 ropes or one degree-3 rope, plus zero or more degree-1 ropes. New degree-1 ropes all have w as their knot. The new degree-2 ropes may have no internal vertices, in which case they are fully placed. Placing w creates a degree-3 rope if w was part of a degree-2 rope but did not lie on the path between its two knots. In this case a new splitter s is identified (marked by a circle in Fig. 11), which will be placed in the next odd placement. The top right case depicted in Fig. 11 shows two dashed arrows, indicating that there are two possible locations for w. Which of the two we will use depends on the matching partner of the splitter s. We explain below how to make this decision.

Odd placement. Before an odd placement, each matching edge has both of its vertices placed, or neither of its vertices placed. There are two cases: the previous even placement left us with a splitter, or not. If there is no splitter,

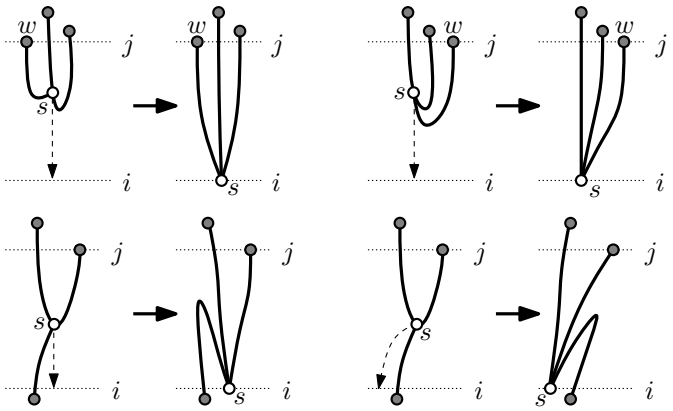

Figure 12: Odd placement for a splitter and a TTT-rope or a TTB-rope.

then we place any unplaced vertex, whose placement does not create a splitter. Any vertex that is directly adjacent to an already placed vertex qualifies. If there is a splitter s, then we place it next. If s is part of a TTT-rope or a TTB-rope, then we place it at the lowest unused y-coordinate i, which creates two or three new TB-ropes and one or zero new BB-ropes, respectively. Symmetrically, if s is part of a TBB-rope or a BBB-rope, then we place it at the highest unused y-coordinate j.

There are two additional things to consider. Let u be the matching partner of the splitter s. By construction u has not been placed yet, but it will be placed in the next even placement, on the same y-coordinate as s.

(1) If s was part of a TTT-rope (or symmetrically, a BBB-rope), then placing s creates three new TB-ropes. If u is part of one of these TB-ropes, then we need to ensure that this particular TB-rope is not in the middle TB-rope. The TTT-rope was created by placing a vertex w at y-coordinate j in the previous step (top right case in Fig. 11). Recall that we had two choices for the location of w. At least one of the two ensures that u is in a rope on the outside (see Fig. 12 (top row)). Hence we now make one of the choices we had in the even placement, and place w accordingly. Placing s might also have created one or more B-ropes. If u is part of one of these B-ropes, then we need to ensure that this particular B-rope is the leftmost or rightmost one that has s as a knot. We can easily do this by ordering the degree 1-ropes with knot s accordingly.

(2) If s was part of a TTB-rope (or symmetrically, a TBB-rope), then placing s creates two TB-ropes and one BB-rope. If u is part of one of the TB-ropes, then we have to ensure again that this particular TB-rope is on the outside. Fortunately, we have a choice of two possible locations for s, see Fig. 12 (bottom row). One of the two ensures that u is in a rope on the outside, hence we place s accordingly. Again, placing s might also have created one or more B-ropes. In general we can place these B-ropes in an arbitrary order. But if u is part of a B-rope, then we need to place this particular rope on the outside.

Placement regions. At each odd and even placement, we have chosen a vertex and assigned a y-coordinate to it. As a result all matching edges are drawn horizontally at different y-coordinates. We have also chosen the left-to-right order of the new ropes this vertex is incident to. To ensure that a straight-line drawing of the tree will be plane, we use a *placement region* for each rope, where all placement regions are disjoint. We choose the placement region for each rope to be a parallelogram bounded by two horizontal lines and two other lines. If i and j are the lowest and highest unused y-coordinates, then the horizontal lines are $y = i$ and $y = j$. The knots of the rope and part of their incident edges are not placed inside the parallelogram, but all unplaced vertices and edges between them will be inside.

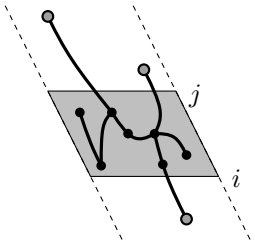

Figure 13: Parallelogram for a TTB-rope.

When a rope is split, we determine a number of disjoint, new parallelograms for the new ropes which are inside the parallelogram of the rope that was split. This ensures that the parallelograms of all ropes after the split are again disjoint.

More precisely, we maintain the following geometric invariant, which holds for every rope after every even placement, see Fig. 13. Let i and j be the lowest and highest unused y-coordinates. There exists a parallelogram between the horizontal lines $y = i$ and $y = j$ and two other lines, such that the knots of the rope lie strictly between the supporting lines of the non-horizontal sides of the parallelogram. This invariant ensures that the rope can be drawn crossing-free with straight lines inside the parallelogram, where only the knots are outside and part of the edges they are incident to. The edges incident to the knots will intersect only the horizontal sides of the parallelogram. Fig. 14 shows the parallelograms for the example of Fig. 10. In the following we argue how to maintain the geometric invariant as the embedding algorithm proceeds.

Consider a rope R and its associated parallelogram P, and assume that the invariant holds after an even placement. The next two steps of the embedding

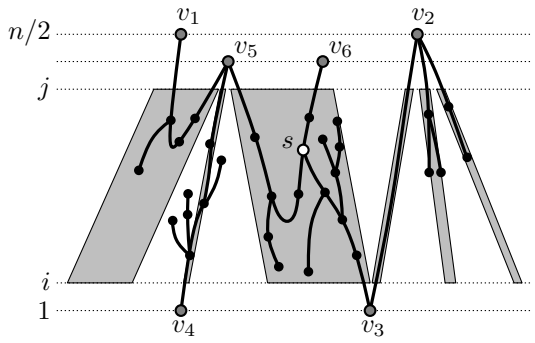

Figure 14: The parallelograms for the ropes of the example in Figure 10.

algorithm place two vertices at the y-coordinate i or j. If neither vertex is part of R then we simply shorten P. However, if one or both of the newly placed vertices was part of R then we need to split P into several disjoint subparallelograms, one for each rope created by the placement. The cases are many, but not difficult (and they are similar to the cases discussed in [3]). Instead of describing all cases in full detail, we restrict ourselves to two representative examples.

The first example shows a vertex of a TT-rope that is placed at y-coordinate i. Fig. 15 illustrates the situation before placement, the situation after placement, and the new parallelograms inside the old one. Note that there is room for any number of new degree-1 ropes. By making the new parallelograms very narrow, we can ensure that they are disjoint at the bottom as well since their lower y-coordinate is $i + 1$.

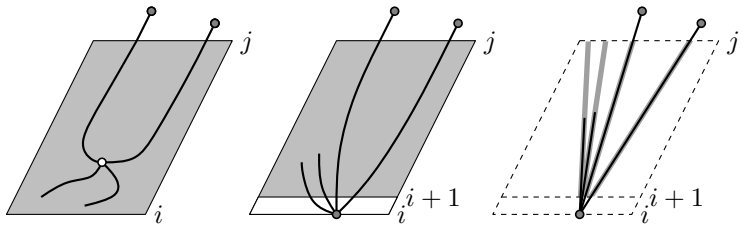

Figure 15: Splitting the parallelogram of a TT-rope.

The second example shows a splitter s that is placed at y-coordinate i, while its matching partner u is contained in one of the B-ropes created by this placement (see Fig. 16). Recall that we place the rope with u on the outside with respect to s, which allows us to place u at y-coordinate i as well. We choose the x-coordinates of s and u so that parallelograms can be found for each new rope within the old parallelogram. Again, by making the new parallelograms very narrow, we can ensure that they are disjoint at the bottom.

Theorem 5 *A tree and a matching always admit a geometric simultaneous embedding.*

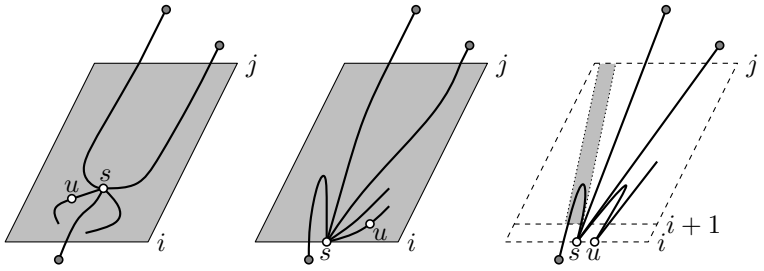

Figure 16: Splitting the parallelogram of a TTB-rope, only one of the five new parallelograms is shown.

7 Many matchings and more

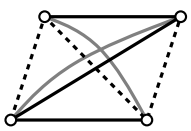

Figure 17: Two paths and a matching.

Geometric simultaneous embeddings have also been considered for more than two graphs. For example, Brass *et al.* [2] show that three paths may not admit a geometric simultaneous embedding. Two paths always admit a geometric simultaneous embedding where one path is drawn in a y-monotone and the other in an x-monotone fashion. An intriguing open question in this context is the following: do two paths and a matching always admit a geometric simultaneous embedding? The simple example in Fig. 17 shows that the matching edges have to have at least two orientations and that the paths cannot both be drawn in a monotone fashion.

The union of two matchings is a set of cycles which can be drawn as rectilinear staircase polygons such that each matching is drawn with either only vertical or horizontal edges. A set of cycles is a subgraph of an outerzigzag and hence by Theorem 3 three matchings always admit a geometric simultaneous embedding. However, it is unclear if one can restrict the set of orientations for each matching to be constant. Furthermore, a set of cycles is also a *maximum-degree-two* graph, that is, a graph such that every vertex has degree at most two. Duncan, Eppstein, and Kobourov [4] proved that two maximum-degree-two graphs always admit a geometric simultaneous embedding. This immediately implies that four matchings also always admit a geometric simultaneous embedding, but again, it is unclear if one can restrict the set of orientations for each matching.

We finally argue that a set of six matchings may not admit a geometric simultaneous embedding. Consider the six vertices of a $K_{3,3}$ to be labeled. Then there are six different ways to take a subset of three edges that form a matching (see Fig. 18). Now consider any pair of edges of the $K_{3,3}$ that are incident to four different vertices. One of the six matchings contains this pair. Therefore, the planarity of all six matchings in a geometric simultaneous embedding implies a planar embedding of $K_{3,3}$. Hence, six matchings may not admit a geometric simultaneous embedding.

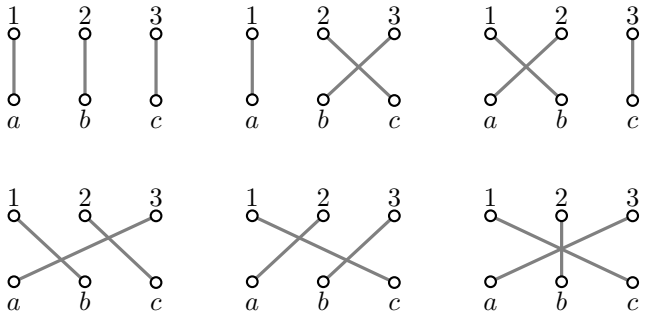

Figure 18: Six matchings that cover $K_{3,3}$.

8 Conclusions and open problems

We presented the first results for geometric simultaneous embeddings where one of the graphs is a matching. Specifically, we showed that there exist planar graphs that do not admit a geometric simultaneous embedding with a matching. We do not know whether this negative result holds also under the additional constraint that the matching and the planar graph do not have any edges in common. Note here that a path and a planar graph might not have a geometric simultaneous embedding, even if they do not share any edges [9].

We also described algorithms that compute a geometric simultaneous embedding of a matching and a wheel, outerpath, or tree. Our drawing algorithms minimize the number of orientations used to draw the edges of the matching. The main remaining open question in this context is: do an outerplanar graph and a matching always admit a geometric simultaneous embedding?

Three or four matchings always admit a geometric simultaneous embedding, but it is unclear if one can restrict the set of orientations for each matching to be constant. Furthermore, the following two questions remain open: do five matchings or two paths and a matching always admit a geometric simultaneous embedding?

Acknowledgments

G. Liotta is supported by MIUR of Italy under project AlgoDEEP prot. 2008TFBWL4. B. Speckmann and K. Verbeek are supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.022.707.

References

[1] P. Angelini, M. Geyer, M. Kaufmann, and D. Neuwirth. On a tree and a path with no geometric simultaneous embedding. In *Proc. 18th International Symposium on Graph Drawing (GD 10)*, 2010. To appear.

[2] P. Brass, E. Cenek, C. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. Kobourov, A. Lubiw, and J. Mitchell. On simultaneous planar graph embeddings. *Computational Geometry: Theory & Applications*, 36(2):117–130, 2007.

[3] E. Di Giacomo, W. Didimo, M. van Kreveld, G. Liotta, and B. Speckmann. Matched drawings of planar graphs. *Journal of Graph Algorithms and Applications*, 13(3):423445, 2009.

[4] C. Duncan, D. Eppstein, and S. Kobourov. The geometric thickness of low degree graphs. In *Proc. 20th ACM Symposium on Computational Geometry*, pages 340–346. ACM, 2004.

[5] C. Erten, P. Harding, S. Kobourov, K. Wampler, and G. Yee. Exploring the computing literature using temporal graph visualization. In *Proc. Conference on Visualization and Data Analysis*, pages 45–56, 2003.

[6] C. Erten, P. Harding, S. Kobourov, K. Wampler, and G. Yee. GraphAEL: Graph animations with evolving layouts. In *Proc. 11th International Symposium on Graph Drawing (GD 03)*, number 2912 in LNCS, pages 98–110. Springer, 2004.

[7] C. Erten and S. Kobourov. Simultaneous embedding of planar graphs with few bends. *Journal of Graph Algorithms and Applications*, 9(3):347–364, 2005.

[8] A. Estrella-Balderrama, E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. Simultaneous geometric graph embeddings. In *Proc. 15th International Symposium on Graph Drawing (GD 07)*, number 4875 in LNCS, pages 280–290. Springer, 2008.

[9] F. Frati, M. Kaufmann, and S. Kobourov. Constrained simultaneous and near-simultaneous embeddings. *Journal of Graph Algorithms and Applications*, 13(3):447–465, 2009.

[10] M. Geyer, M. Kaufmann, and I. Vrt'o. Two trees which are self-intersecting when drawn simultaneously. *Discrete Mathematics*, 309(7):1909–1916, 2009.

[11] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002.