# On Visibility Representations of Non-Planar Graphs*

**Therese Biedl†1, Giuseppe Liotta‡2, and Fabrizio Montecchiani§3**

1 **David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada**
`biedl@uwaterloo.ca`

2 **Dipartimento di Ingegneria, Università degli Studi di Perugia, Perugia, Italy**
`giuseppe.liotta@unipg.it`

3 **Dipartimento di Ingegneria, Università degli Studi di Perugia, Perugia, Italy**
`fabrizio.montecchiani@unipg.it`

## Abstract

A rectangle visibility representation (RVR) of a graph consists of an assignment of axis-aligned rectangles to vertices such that for every edge there exists a horizontal or vertical line of sight between the rectangles assigned to its endpoints. Testing whether a graph has an RVR is known to be NP-hard. In this paper, we study the problem of finding an RVR under the assumption that an embedding in the plane of the input graph is fixed and we are looking for an RVR that reflects this embedding. We show that in this case the problem can be solved in polynomial time for general embedded graphs and in linear time for 1-plane graphs (i.e., embedded graphs having at most one crossing per edge). The linear time algorithm uses a precise list of forbidden configurations, which extends the set known for straight-line drawings of 1-plane graphs. These forbidden configurations can be tested for in linear time, and so in linear time we can test whether a 1-plane graph has an RVR and either compute such a representation or report a negative witness. Finally, we discuss some extensions of our study to the case when the embedding is not fixed but the RVR can have at most one crossing per edge.

**1998 ACM Subject Classification** I.3.5 Computational Geometry and Object Modeling

**Keywords and phrases** Visibility Representations, 1-Planarity, Recognition Algorithm, Forbidden Configuration

**Digital Object Identifier** 10.4230/LIPIcs.SoCG.2016.19

## 1 Introduction

A visibility representation is an appealing method of displaying graphs. It consists of assigning axis-aligned rectangles to vertices and horizontal or vertical line segments to edges in such a way that line segments of edges begin and end at the rectangles representing their endpoints and intersect no rectangles in-between. Equivalently, each edge corresponds to a horizontal or

[*] Because of space limitations some proofs are only sketched or omitted; complete proofs can be found in the full version of the paper [2].

[†] Research of TB supported by NSERC.

[‡] Research of GL supported in part by the MIUR project AMANDA "Algorithmics for MAssive and Networked DAta", prot. 2012C4E3KT_001.

[§] Research of GL and FM supported in part by the MIUR project AMANDA "Algorithmics for MAssive and Networked DAta", prot. 2012C4E3KT_001. Research undertaken while FM was visiting the University of Waterloo.

32nd International Symposium on Computational Geometry (SoCG 2016).
Editors: Sándor Fekete and Anna Lubiw; Article No. 19; pp. 19:1–19:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

vertical line-of-sight between its endpoints; hence the name "visibility". Edge segments may cross each other, but any such crossing occurs at a right angle. These right-angle crossings as well as the absence of bends leads to a high readability of the visualization. See Fig. 1.

Visibility representations were introduced to the Computational Geometry community in 1985, when Wismath [26] showed that every planar graph has a bar-visibility representation, i.e., where vertices are represented as horizontal bars and edges correspond to vertical visibilities. (The same result was discovered independently multiple times [10, 17, 19, 23, 24].)

Such bar-visibility representations can exist only for planar graphs, so for representing non-planar graphs one must use edge segments in both directions. (We use the term *rectangle visibility representation* or *RVR* to distinguish such drawings from bar-visibility representations.) Clearly not any graph $G$ can have an RVR either; for example $G$ must have thickness 2 (i.e., be the union of two planar graphs), and, as was shown in [4], it must have at most $6n - 20$ edges. But neither of these conditions is sufficient. In fact, Shermer showed in 1996 that it is NP-hard to test whether a given graph has an RVR [20].

In this paper, we study the problem of testing whether a graph has a rectangle visibility representation if some aspects of it are fixed a priori. In previous work by Streinu and Whitesides [21], it was shown that testing whether a graph has an RVR is polynomial if the following additional information is given: we know for each edge whether it is drawn horizontal or vertical, we know the rotational scheme (i.e., the fixed order of edges around each vertex), and we know which vertices form the outer face of the visibility representation. Streinu and Whitesides give necessary and sufficient conditions for when such restrictions can be realized, and show that these can be tested in $O(n^2)$ time.

In this paper, we study a slightly different scenario. Like Streinu and Whitesides, we assume that the rotational scheme and the outer face is fixed. However, we do not require to know the directions of the edges; instead we must know a priori which edges cross each other, and the order in which these crossings occur along each edge (if there is more than one crossing on an edge). In other words, the graph has a *fixed embedding* (see also Section 2). A reduction to the topology-shape-metrics approach then allows to test in $O(n^{1.5} \log n)$ time whether a graph with $n$ vertices has a rectangle visibility representation; see Section 2.2.

As our main result, we then consider the case when any edge contains at most one crossing (the so-called *1-planar graphs*). Previous attempts to create visibility representations for 1-planar graphs had relaxed the restriction that edges must not cross non-incident vertices, and defined a *bar $k$-visibility drawing* to be a bar-visibility drawing where each line of sight can intersect at most $k$ bars [8]. In other words, each edge can cross at most $k$ vertices. These crossings are arguably less intuitive than crossings between edges. Brandenburg *et al.* [5] and independently Evans *et al.* [11] prove that 1-planar graphs have a bar 1-visibility drawing (if we are allowed to change the embedding).

In contrast to these results, we consider in Section 3 embedding-preserving rectangle visibility representations of embedded 1-planar graphs, also called 1-plane graphs. It is easy to see that not all 1-plane graphs have such an RVR (see also Section 4). The main contribution of our paper is to show that for a 1-plane graph it is possible to test in linear time whether it has an RVR. The approach is entirely different from the topology-shape-metrics above and reveals much insight into the structure required for an RVR to exist. Specifically, we consider three configurations (called B-configuration, W-configuration, and the newly defined T-configuration) and show that a 1-plane graph has an RVR if and only if none of these three configurations occurs. This mirrors in a pleasing way the result by Thomassen [25], which shows that a 1-plane graph has a straight-line drawing if and only if it contains no B-configuration or W-configuration. Our result also provides an answer (in a special case) to Shermer's question of characterizing when graphs have RVRs [20]. Also, we prove that testing whether a 1-plane graph contains any of the three forbidden configurations can be done in linear time, and in the absence of them we can compute in linear time an RVR which fits into a grid of quadratic size. It may be worth recalling that embedding-preserving straight-line drawings of 1-plane graphs may require exponential area [13]. In Section 4 we show both negative and positive results in the variable embedding setting. Finally, in Section 5 we conclude with a discussion and open problems.

## 2 Definitions and Basic Results

We only consider *simple graphs*, i.e., graphs with neither loops nor multiple edges. A *drawing* $\Gamma$ of a graph $G$ maps each vertex to a point of the plane and each edge to a Jordan arc between its two endpoints. We only consider *simple drawings*, i.e., drawings such that the arcs representing two edges have at most one point in common, which is either a common endpoint or a common interior point where the two arcs properly cross. A drawing is *planar* if no two arcs cross. A drawing divides the plane into topologically connected regions, called *faces*. The unbounded region is called the *outer face*. For a planar drawing the boundary of a face consists of vertices and edges, while for a non-planar drawing the boundary of a face may contain vertices, crossings, and edges (or parts of edges). An *inner* face/edge/vertex is a face/edge/vertex that is not (part of) the outer face. A *rotational scheme* of a graph $G$ consists of assigning at each vertex a cyclic order of the edges incident to that vertex. A drawing is said to respect a rotational scheme if scanning around the vertex in clockwise order encounters the edges in the prescribed order. We say that a graph has a *fixed embedding* if the graph comes with a rotational scheme, a fixed order of crossings along each edge, and with one face indicated to be the outer face. (One can show that fixing the rotational scheme and the order of crossings fixes all faces.) Given a graph $G$ with a fixed embedding, the *planarization* $G_p$ of $G$ is the graph obtained by replacing each crossing of $G$ with a *dummy-vertex*. The vertices of $G$ in $G_p$ are called *original vertices* to avoid confusion. The planarization $G_p$ is a *plane* graph, i.e., a planar graph with a fixed embedding without crossing.

A *visibility representation* (also called *rectangle visibility representation* or RVR for short) of a graph $G$ is an assignment of disjoint axis-aligned rectangles to the vertices in such a way that for every edge there exists a horizontal or vertical line of sight between its endpoints (not all such line of sights give rise to an edge). We follow a commonly adopted model where the lines of sight are *thick*, i.e., have non-zero area (see e.g. [16, 21, 23, 26]). Thus we may assume that each edge is routed as a (horizontal or vertical) line segment, attaching at each end at a point that is not a corner of the rectangle of that vertex. In what follows, when this leads to no confusion, we shall use the same term *edge* to indicate both an edge of a graph, the line of sight between two rectangles, and the line segment representing the edge, and we use the same term *vertex* for both the vertex of a graph and the rectangle that represents it. We observe that if we have an RVR, then we can naturally extract a drawing from it as follows. Place a point for each vertex $v$ inside the rectangle representing $v$ and connect it to all the attachment points of incident edges of $v$ inside the rectangle; this adds no crossing. We say that an RVR $\Gamma$ of a graph $G$ *respects* the embedding of $G$ if applying this operation to $\Gamma$ results in the same embedding.

### 2.1 Topology-shape-metrics

The *topology-shape-metrics* approach is a well-known method introduced by Tamassia [22] to create orthogonal drawings of graphs, i.e., drawings where all vertices are points and edges

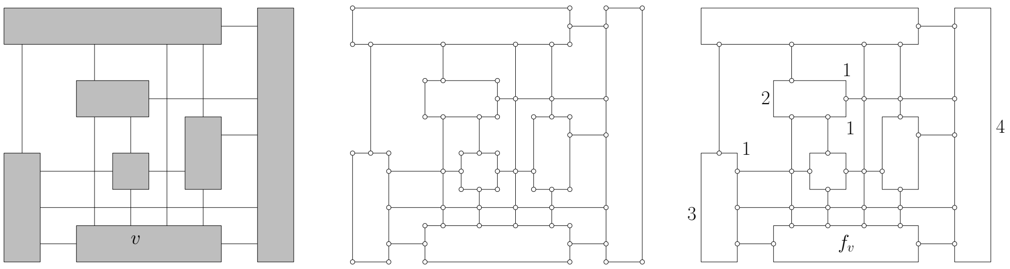


**Figure 1** A visibility representation of a graph, the orthogonal representation obtained from it, and the orthogonal representation used to compute it. We show only selected non-zero bend-counts.

are curves with horizontal and vertical segments. In this section, we briefly review how any RVR can be expressed as an orthogonal-shape-metric; this is quite straightforward but has to our knowledge not been used before. We will use such metrics twice in our paper: once to determine whether a given embedded graph has an RVR, and once to store an RVR obtained for 1-plane graphs so that they can be manipulated efficiently.

Consider an RVR $\mathcal{R}$ and interpret it as a PSLG, i.e., a planar straight-line graph in the computational geometry sense, by replacing corners, attachment points and crossings by nodes as in Fig. 1. We get a planar graph that has special properties:

▶ **Definition 1.** An *orthogonal representation* is a plane graph $H$ with maximum degree 4, an assignment of an angle $\alpha(v, f) \in \{90°, 180°, 270°\}$ to any vertex-face-incidence (between vertex $v$ and face $f$) of $H$, and a bend-count $b(e, f) \in \{0, +1, -1, +2, -2, \dots\}$ to any edge-face-incidence (between edge $e$ and face $f$) of $H$, such that the following is satisfied:

- at every vertex $v$, the angles at $v$ sum to 360°,
- at every edge $e$, the two bend-counts sum to 0,
- for every inner face $f$ with $k$ vertices, the angles and bend-counts (multiplied by 90°) at $f$ sum to $(k-2)180°$.
- the angles and bend-counts (multiplied by 90°) at the outer face sum to $(k+2)180°$, where $k$ is the number of vertices on the outer face.

▶ **Observation 2.** *If $G$ has an RVR $\Gamma$, then construct a plane graph $H$ by replacing every crossing, every edge-vertex attachment point and every corner of a vertex-rectangle of $\Gamma$ with a node in $H$. The resulting graph then has an orthogonal representation where all bend-counts are 0.*

Tamassia showed [22] that for any orthogonal representation, one can reconstruct a planar orthogonal drawing of the graph that respects these angles and bend-counts. Tamassia also showed how to compute for a given plane graph $H$ some angle-assignment and bend-counts such that the result is an orthogonal representation. This is done via a flow-approach; see [22] or [9] for details. This approach allows also to specify upper and lower bounds on angles and edge-counts.

## 2.2 Testing embedded graphs for RVRs

Now presume that we are given a graph $G$ with a fixed embedding. In this section, we will show how to test whether $G$ has an RVR that respects its embedding using the topology-shape-metrics approach. Note that the embedding tells us nearly the entire graph $H$ of the orthogonal representation of a (putative) RVR of $G$. The only thing unknown is the location of the corners of each rectangle of a vertex. We can now interpret these corners as bends (rather than as nodes) in the topology. Hence define a graph $H$ as follows (see also Fig. 1):

- Start with the planarization $G_p$ of $G$.
- For any original vertex $v$ of $G$, define a cycle $C_v$ with $\deg(v)$ nodes in $H$. Attach the (parts of) edges incident to $v$ to the nodes of cycle $C_v$ in the specified order of the planar embedding, in such a way that the interior of the cycle forms a face. In other words, form a cycle that can represent the boundary of the rectangle of $v$.
- Impose an upper and lower bound of 0 onto the bend-counts of all (parts of) edges of $G$. In other words, we do not allow any bends for the original edges.
- Impose a lower bound of 0 onto each bend-count $b(e, f_v)$ where $e$ is an edge of some cycle $C_v$ and $f_v$ is the face formed by cycle $C_v$. In other words, cycle $C_v$ may have bends, but all such bends must form 90° angles towards $f_v$.
- If $a$ is a node of $C_v$, then impose an upper and lower bound of 180° onto the angle $\alpha(a, f_v)$, where $f_v$ as before is the face inside $C_v$. In other words, at attachment points of edges face $f_v$ must not have a corner.

▶ **Theorem 3.** *Let $G$ be a with a fixed embedding and $n$ vertices and crossings. Then we can test in $O(n^{1.5}\log n)$ time whether $G$ has an RVR that respects this embedding.*

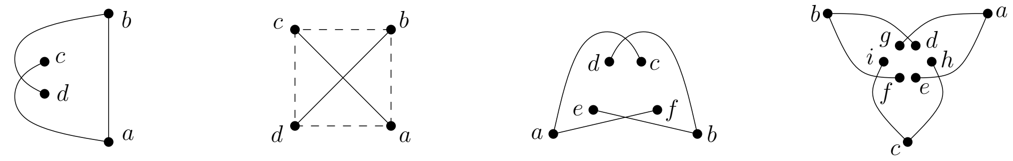
**Proof.** Build the graph $H$ as described above. It should be obvious that if we can find an orthogonal representation of $H$ that respects these constraints, then it gives rise to an RVR of $G$. Indeed, for any vertex $v$ the cycle $C_v$ must form a rectangle: It cannot have a bend at any attachment-point, and any bend that it has provides an angle of 90°, hence by the condition it has exactly four such bends. At any dummy-vertex the four angles must be 90° since its degree is four. All edges of $H$ that came from edges of $G$ are drawn straight, and continue straight at any crossing, so give rise to a line of sight. We then use Tamassia's flow-approach to test whether an orthogonal representation of $H$ exists that satisfies all constraints. This takes $O(n^{1.5}\log n)$ time [7]. ◀

# 3 Embedded 1-planar graphs

While the topology-shape-metrics approach gives a polynomial way to test whether a graph has an RVR, its black-box approach is somewhat unsatisfactory. In particular, in case of a negative answer, we obtain no good insights as to which parts of the graph prevented the existence of a representation. In this section, we therefore turn our attention to 1-planar graphs. We chose this graph class for two reasons. One is that they are known to have thickness 2 and at most $4n - 8$ edges [18], and so they are good candidates for always having an RVR. (We will, however, see that this is not the case.) Secondly, 1-planar graphs have been studied widely in the graph theory and graph drawing communities (see e.g. [1, 5, 11, 13, 18, 25]), and visibility representations of 1-planar graphs hence should be of interest.

## 3.1 Background

A graph is *1-planar* if it has a drawing with at most one crossing per edge. A *1-plane* graph is a 1-planar graph with a fixed embedding. Recall that this means that we have a fixed order of edges at each vertex, it is fixed which pairs of edges cross, and we know what the faces are and which one of them is the outer face. A 1-plane graph $G$ is called *triangulated 1-plane* if all faces are triangles. In a 1-plane graph, no two crossings can be adjacent and so a triangle is composed of three vertices or two vertices and a crossing point.



**(a)** B **(b)** Kite **(c)** W **(d)** T

**Figure 2** Possible crossing configurations in a 1-plane graph $G$. (a) A B-configuration. (b) A (subgraph) of a kite. The dashed edges either do not exist or define faces. (c) A W-configuration. (d) A T-configuration.

Let $e = (a, c)$ and $e' = (b, d)$ be two edges of $G$ that cross at a point $p$. We say that $e, e'$ induce a *B-configuration* if there exists an edge between their endpoints (say edge $(a, b)$) such that (in the induced drawing of $\{a, b, c, d\}$) the triangle $\{a, b, p\}$ contains vertices $c$ and $d$ inside (see also Fig. 2a).

If the crossing is not in a B-configuration, then there are two further possibilities for edge $(a, b)$. The triangle $(a, b), (b, p), (p, a)$ may be an interior face, or it could have other vertices inside. If, for all pairs in $\{a, c\} \times \{b, d\}$, there either is no edge between the vertices or the triangle that it forms is a face, then we could add the edges that did not exist previously, routing them along the crossing, and then obtain a *kite*, i.e., a crossing with a 4-cycle among its endpoints such that removing the crossing turns the 4-cycle into a face.

Let $(a, c)$ and $(b, d)$ be two edges of $G$ that cross at a point $p$, and let $(a, f)$ and $(b, e)$ be two further edges of $G$ that cross at a point $q$. The four edges induce a *W-configuration* if vertices $c, d, e, f$ lie inside the closed region delimited by the edge segments $(a, p)$, $(b, p)$, $(a, q)$, and $(b, q)$ (see also Fig. 2c).

We introduce a fourth configuration, called the *trillium configuration*, or *T-configuration* for short, which is illustrated in Fig. 2d. Namely, let $(a, c)$ and $(b, d)$ be a pair of edges of $G$ that cross at a point $p$. Let $(a, e)$ and $(c, h)$ be a second pair of edges that cross at a point $q$. Let $(b, f)$ and $(c, i)$ be a third pair of edges that cross at a point $t$. The six edges induce a T-configuration if vertices $d$, $e$, $f$, $g$, $h$, $i$ lie inside the closed region delimited by the edge segments $a - p - b - t - c - q - a$. Vertices $a, b, c$ are called the *outer vertices* of the T-configuration, while the remaining six vertices are the *inner vertices*.

## 3.2 Main result and outline

In this section we prove the following characterization of 1-plane graphs that admit an RVR:

▶ **Theorem 4.** *A 1-plane graph $G$ admits an RVR if and only if it contains no B-configuration, no W-configuration, and no T-configuration.*

Notably, Theorem 4 extends the characterization for straight-line drawability of 1-plane graphs given by Thomassen [24], adding the T-configuration to the set of obstructions. We prove necessity in Section 3.3 and sufficiency in Section 3.4. The latter comes with an efficient algorithm to construct an RVR $\Gamma$ of a drawable graph $G$. Since testing for forbidden configurations can also be done efficiently (Section 3.5), we get:

▶ **Theorem 5.** *Let $G$ be a 1-plane graph with $n$ vertices. There exists an $O(n)$-time algorithm to test whether $G$ admits an RVR. Also, in the positive case the algorithm computes an RVR of $G$ that has $O(n^2)$ area.*

## 3.3 Proof of necessary condition

To prove the necessity of Theorem 4, we show a slightly stronger statement:

▶ **Lemma 6.** *Let $G$ be a graph that admits an RVR $\Gamma$ and whose outer face is a cycle $\mathcal{C}$. If while walking along $\mathcal{C}$ we encounter $k$ vertices and $c$ crossing points, then $k \geq 3$ and $c \leq 2k - 4$.*

**Proof.** Convert $\Gamma$ into an orthogonal representation (Observation 2) and consider the angles at the outer face. Since there are $k$ vertices on the outer face, we have $2k$ edge-attachment points, and each contributes 90°. Each of the $c$ crossings contributes 90°. If $b$ is the number of rectangle-corners on the outer face, then $b \leq 4k$, and each contributes 270°. In total the outer face has $2k+c+b$ nodes and the sum of angles must be $180°(2k+c+b+2)$. But we also know that the angles sum to $90°(2k+c)+270°b = 180°(2k+c+b+2)+90°(b-2k-c-4)$. So $b-2k-c-4=0$, hence $4k \geq b = 2k+c+4$, hence $c \leq 2k-4$. This implies $k \geq 2$, and in fact $k \geq 3$ is required, else $c = 0$ and the outer face would consist of a double edge. ◀

Now assume that graph $G$ a B-configuration (W-configuration, T-configuration), and consider the subgraph $G'$ induced by that configuration. Then the outer face of this subgraph is a cycle with $k = 2$ and $c = 1$ ($k = 2$ and $c = 2$, $k = 3$ and $c = 3$). So $G'$ (and with it, $G$) cannot have an RVR and the necessity holds.

## 3.4 Proof of sufficient condition

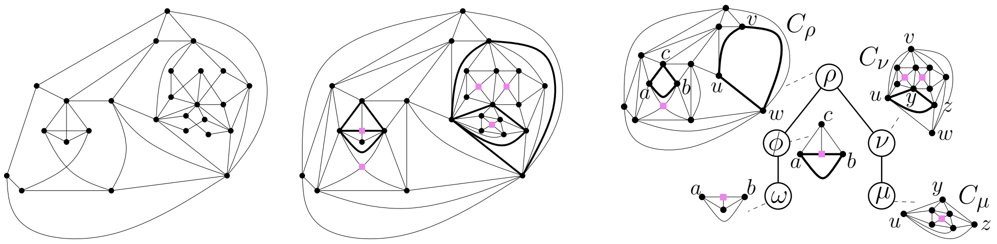
We show that the absence of any B-/W-/T-configuration suffices to compute an RVR. Let $G$ be a 1-plane graph with $n$ vertices and with no B-/W-/T-configuration as a subgraph, see also Fig. 3a for an example. We describe a drawing algorithm, `1PRVDrawer`, which computes an RVR $\Gamma$ of $G$. We assume that $G$ is 3-connected (Sections 3.4.1–3.4.6); an extension to the general case is proved in the full-length version [2].

### 3.4.1 Triangulating and planarizing

The first step of algorithm `1PRVDrawer` is to triangulate $G$. This is a well-known operation for 3-connected 1-plane graphs, see [1, 13]. However, these algorithms all modify the given 1-planar embedding, even if there is no B-configuration or W-configuration. In fact, this may be required since every crossing must form a kite in a triangulation of $G$, and so if an edge between endpoints of a crossing already existed, but did not form a face, then it must be re-routed. We do not want to change the embedding, and hence cannot triangulate the 1-plane graph per se. Instead, we combine the triangulation step with a planarization step to ensure that the result has no multiple edges, yet at all crossings (respectively the corresponding dummy-vertices) we have kites.

We construct the triangulated planar graph $G^+$ with the following steps:

- Let $G_1$ be the planarization of $G$. Note that dummy-vertices have degree 4 (and we will maintain this throughout later modifications).
- For each dummy-vertex $z$ of $G_1$, let $v_0, \ldots, v_3$ be the four neighbors in clockwise order. For $i = 0, \ldots, 3$, consider the face at $z$ between the edges $(z, v_i)$ and $(z, v_{i+1})$ (addition modulo 4). If this face is the outer face, or an inner face that is not a triangle, then add the edge $(v_i, v_{i+1})$. Put differently, we add the edges that are needed to turn $z$ into a kite. (In what follows, we will use the previously defined terms "kite", "B-configuration", "W-configuration" and "T-configuration" for the planarized version as well, i.e., for the



**(a)** $G$ **(b)** $G^+$ **(c)** $\mathcal{T}$

**Figure 3** (a) A 1-plane graph $G$. (b) The triangulated planar graph $G^+$ obtained from $G$. Dummy-vertices are pink squares. (c) The 4-block tree $\mathcal{T}$ of $G^+$.

same situations with crossings replaced by dummy-vertices.) Call the resulting graph $G_2$. The following result (proved in [2]) is crucial for our correctness:

▶ **Lemma 7.** *If $G$ is simple and 3-connected and has no B-configuration, W-configuration or T-configuration, then $G_2$ has no B-configuration, W-configuration, or T-configuration and no multiple edges.*

- In the plane graph $G_2$ each dummy-vertex is surrounded by a kite by construction. Hence all faces of $G_2$ that are not triangles contain only original vertices. Triangulate each such face arbitrarily; this does not create a B-configuration or multiple edges since in a planar graph this would imply a cutting pair.

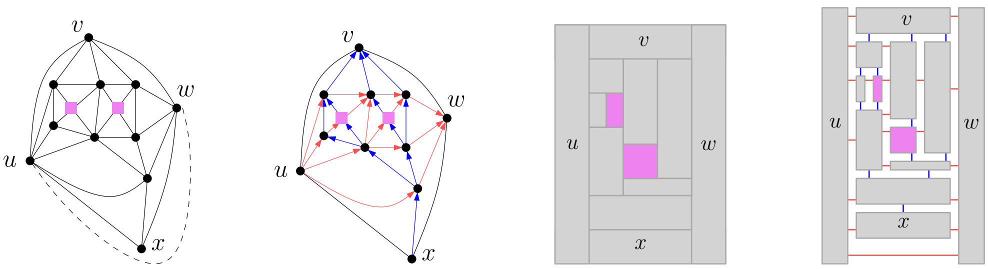
The resulting graph $G^+$ is the plane triangulated graph that we use to obtain the RVR. By construction it has the following crucial properties: It is simple, has no B-configuration, W-configuration or T-configuration, every dummy-vertex has degree 4, and removing from $G^+$ all the edges inserted in the above procedure, and replacing each dummy-vertex with a crossing point, we obtain the original 1-plane graph $G$.

### 3.4.2 The 4-block tree and an outline

Algorithm `1PRVDrawer` uses a decomposition of $G^+$ into its 4-connected components, i.e., a 4-connected triangulated subgraph of $G^+$. Let the *4-block tree* $\mathcal{T}$ of $G^+$ be a tree defined as follows (see also [14] and Fig. 3c). Every 4-connected component $C_\nu$ of $G^+$ is represented by a node $\nu$ in $\mathcal{T}$. There is an edge between two nodes $\nu$ and $\mu$ in $\mathcal{T}$, if there is a *separating triangle* (i.e., a triangle with vertices both inside and outside) that belongs to both $C_\nu$ and $C_\mu$. We root $\mathcal{T}$ at the node $\rho$ with the 4-connected component that contains the outer face. Then for any parent $\nu$ and child $\mu$, the separating triangle common to $C_\nu$ and $C_\mu$ is an inner face in $C_\nu$ and the outer face of $C_\mu$. It is known that $\mathcal{T}$ can be computed in $O(n)$ time [14].

We now give an overview of the remaining steps of algorithm `1PRVDrawer`. It first visits $\mathcal{T}$ top-down and determines for each 4-connected component $C_\nu$ a special edge, called *surround-edge*. It also computes an RVR $\gamma_\nu$ of $C_\nu$ using a so-called rectangular dual representation. Next, the algorithm visits $\mathcal{T}$ bottom-up. Let $\nu$ be a node of $\mathcal{T}$ with $h \geq 1$ children $\mu_1, \dots, \mu_h$. Denote by $G_\nu$ the graph whose 4-block tree is the subtree of $\mathcal{T}$ rooted at $\nu$. The already computed visibility representations $\Gamma_{\mu_1}, \dots, \Gamma_{\mu_h}$ of $G_{\mu_1}, \dots, G_{\mu_h}$ are suitably merged into $\gamma_\nu$ and an RVR $\Gamma_\nu$ of $G_\nu$ is obtained.

At the root we obtain an RVR of $G_\rho = G^+$. In order to turn this in an RVR of $G$, we must un-planarize, i.e., replace every dummy-vertex by a crossing. For this to be feasible, we need each dummy-vertex to be drawn in a special way: its rectangle needs to have one incident



**(a)** $C_\nu^-$ **(b)** transv.pair **(c)** $\mathcal{R}$ **(d)** $\mathcal{R}_\nu$

**Figure 4** (a) The graph $C_\nu^-$ obtained from $C_\nu$ in Fig. 3c. (b) A transversal pair of bipolar orientations for $C_\nu^-$. (c) The corresponding rectangular dual representation. (d) Retracting the rectangles and adding the surround-edge.

edge on each side. We call this the *4-sides-condition*. Even then converting dummy-vertices into crossings is non-trivial, since the four edges on the four sides may not be suitably aligned. To achieve such an alignment, we use the so-called "zig-zag-bend-elimination-slide" [3], with which we can transform parts of the drawing until edges are aligned and hence each dummy-vertex can be turned into a crossing point. Since this introduces no other crossings we obtain an RVR, and deleting the added edges gives the desired RVR of $G$.

### 3.4.3 Surround-edges and drawing 4-connected components

Consider a 4-connected component $C_\nu$. We want to find an RVR of $C_\nu$ that satisfies the 4-sides-condition. To be able to merge later, it must satisfy one more property: We pick a so-called *surround-edge* $e$ on the outer face beforehand, and we want an RVR such that all edges incident to an endpoint of $e$ are drawn horizontally.

▶ **Observation 8.** *If $G^+$ has no B-configuration, W-configuration or T-configuration, then there exists an edge $e$ on the outer face $\{u, v, w\}$ of $C_\nu$ such that the vertex facing $e$ from inside $C_\nu$ is an original vertex. Here, the* vertex facing $e$ from the inside *is the third vertex on the face $f_e$ of $G^+$ that is incident to $e$ and inside $\{u, v, w\}$.*

If more than one edge of the outer face of $C_\nu$ satisfies the condition of Observation 8, then we chose the surround-edge arbitrarily with one exception: If $\nu$ has a parent $\pi$, and the surround-edge of $C_\pi$ is one of the outer-face edges of $C_\nu$, then we use the same edge also as surround-edge of $C_\nu$. Note that whether an edge can be used as surround-edge depends on its incident face in $G^+$ (not $C_\pi$), and so the surround-edge for $C_\pi$, if it exists in $C_\nu$, also qualifies as surround-edge for $C_\nu$. For example, for graph $C_\phi$ in Fig. 3c, edge $(a, b)$ must be chosen as surround-edge, because both $(b, c)$ and $(a, c)$ have a dummy-vertex as third vertex on the face. Since edge $(a, b)$ also belongs to $C_\omega$, it becomes the surround-edge of $C_\omega$ as well.

▶ **Lemma 9.** *$C_\nu$ has an RVR such that every inner vertex of degree 4 satisfies the 4-sides-condition and all edges incident to an endpoint of the surround-edge of $C_\nu$ are drawn horizontally.*

**Proof Sketch.** If $C_\nu$ is $K_4$, then such an RVR is easily obtained (see also Fig. 7b), so we assume that $C_\nu$ has at least 5 vertices. Remove the surround-edge from $C_\nu$ to obtain graph $C_\nu^-$. Since $C_\nu$ is 4-connected, $C_\nu^-$ is internally 4-connected. Such a graph is known to have a *transversal pair of bipolar orientations*, which directs and colors each inner edge either

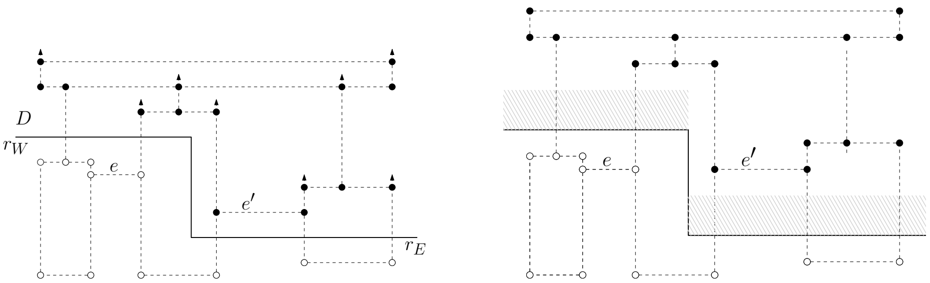


**Figure 5** A zig-zag-slide (adapted from [3]). Black points move upward while white points remain stationary. With this amount of sliding, edges $e$ and $e'$ become aligned.

red or blue[1] such that the following holds: $(i)$ For each inner vertex $v$, the clockwise order of edges around $v$ contains outgoing blue edges, then outgoing red edges, then incoming blue edges, then incoming red edges, and none of these sets is empty. $(ii)$ If $v_N$, $v_E$, $v_S$, $v_W$ are the four vertices on the outer face in clockwise order, then all inner edges incident to $v_N/v_E/v_S/v_W$ are blue incoming/red incoming/blue outgoing/red outgoing. See Fig. 4b. Such a transversal pair of bipolar orientations always exists and can be computed in linear time [12, 15]. Their proof allows us to specify $v_E$ and $v_W$ to be the ends of the surround-edge, and to color all outer face edges red. In the same papers it was also shown that from this coloring and orientation, we can obtain a *rectangular dual representation*, i.e., an assignment of interior-disjoint rectangles to vertices such that the union of these rectangles is a rectangle without holes. For each edge $(v, w)$, the rectangles of $v$ and $w$ share a positive-length part of their boundaries. Moreover, if the edge is directed $v \to w$, then the rectangle of $v$ is left (below) $w$ if and only if the edge is red (blue). See Fig. 4c.

We can now obtain an RVR of $C'_\nu$ easily: use the rectangular dual representation of $C'_\nu$, and retract the rectangles a bit. The 4-sides-condition holds since rectangles of inner vertices have neighbors on all sides. Since outer face edges are red and the surround-edge was $(v_W, v_E)$, all incident edges of $v_W$ and $v_E$ are horizontal. Finally we add the surround-edge after lengthening the rectangles of $v_W, v_E$ and obtain the desired RVR. See Fig. 4d. ◀

### 3.4.4 Zig-zag-slides

Both for the merging step and for un-doing the planarization, we need a method of modifying the drawing such that some items are moved while others are stationary. This can be achieved with a zig-zag-bend-elimination-slide (or zig-zag-slide for short) [3]. Assume that $\Gamma$ is an RVR and we have a dividing curve $D$ as follows: $D$ consists of some vertical line segment $s$ that intersects no horizontal element of the drawing (i.e., neither a horizontal edge nor a horizontal boundary of a rectangle). At the top end of $s$, attach a leftward horizontal ray $r_W$. At the bottom end of $s$, attach a rightward horizontal ray $r_E$. No conditions are being put onto $r_W$ and $r_E$. Define the *region above* $D$ be all points that are in the $x$-range of $r_W$ and strictly above $r_W$, and all points in the $x$-range of $r_E$ and on or above $r_E$.

Now slide all points in the region above $D$ upwards by some amount $\delta > 0$. This maintains a visibility representation, after lengthening vertical edges and rectangle borders, since no horizontal segments cross $s$. See Fig. 5. Similar define zig-zag-slides for the three other shapes of $D$ achieved by flipping the picture horizontally and/or rotating 90°.

[1] In all figures of the paper red edges appear in lighter gray than blue edges when printed b/w.

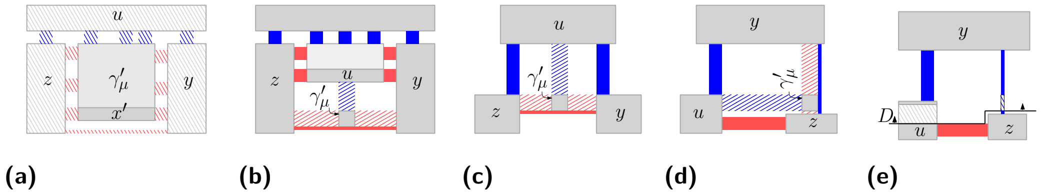


**Figure 6** (a) Illustration of $\gamma_\mu$ and $\gamma'_\mu$. (b) Illustration for the proof of Lemma 10. (c–e) Illustration for the proof of Lemma 11. (c) $u$ has two vertical edges. (d) $y$ has two vertical edges, $u$ has a higher top than $z$. (e) Slide so that $u$ has a higher top than $z$.

### 3.4.5 Merging components

Let $\mu$ be a child of $\nu$ in $\mathcal{T}$, and recall that we already obtained drawings $\gamma_\nu$ of $C_\nu$ and (recursively) $\gamma_\mu$ of the graph $G_\mu$ consisting of $C_\mu$ and the components at its descendants. The common vertices $\{u, y, z\}$ of $G_\mu$ and $C_\nu$ form the outer face of $G_\mu$. See Fig. 3c. The outer face of $C^-_\mu$, consists of $\{u, y, z\}$ as well as a fourth vertex, say $x'$; after possible renaming of $\{u, y, z\}$ we may assume that the order around the outer face of $C^-_\mu$ is $u, y, x', z$. In particular, drawing $\gamma_\mu$ contains (up to rotation) node $u$ on the top, node $y$ on the right, the surround-edge $(y, z)$ and node $x'$ at the bottom, and node $z$ on the left. Let $\gamma'_\mu$ be the drawing obtained from $\gamma_\mu$ by deleting $u, y, z$. It suffices to merge $\gamma'_\mu$, since $u, y, z$ and the edges between them are represented in $\gamma_\nu$ already. See Fig. 6a.

Triangle $\{u, y, z\}$ is an inner face of $C_\nu$. The challenge is now to find a region inside where this face is drawn in $\gamma_\nu$ into which $\gamma'_\mu$ will “fit”. Put differently, we want to find a region within $\gamma_\nu$ inside the face $\{u, y, z\}$ that is (after possible rotation) below $u$, to the left of $y$ and to the right of $z$. We call such a region, a *feasible region* for $\gamma_\mu$ in $\gamma_\nu$. As we have no control over how triangle $\{u, y, z\}$ is drawn in $\gamma_\nu$, the existence of a feasible region is non-trivial, and may in fact require modifying $\gamma_\nu$ slightly. We start with a special case that will be needed later:

▶ **Lemma 10.** *Assume that the surround-edge of $C_\nu$ also belongs to $C_\mu$. Then there exists a feasible region for $\gamma_\mu$ in $\gamma_\nu$.*

**Proof.** If the surround-edge $e$ of $C_\nu$ belongs to $C_\mu$, then by our choice of surround-edge we also used $e$ as the surround-edge for $C_\mu$. That is, $(y, z)$ is the surround-edge for both $C_\mu$ and $C_\nu$. In $\gamma_\nu$, $(y, z)$ was drawn (after possible rotation) bottommost, with $u$ above it. A feasible region is now found in the small strip above the drawing of edge $(y, z)$. See Fig. 6b. ◀

▶ **Lemma 11.** *A feasible region for $\gamma_\mu$ in $\gamma_\nu$ always exists, possibly after modifying $\gamma_\nu$ without changing angles or incidences.*

**Proof.** We already argued this for the case where $\{u, y, z\}$ includes the surround-edge $e$ of $C_\nu$, so assume that this is not the case. Then face $\{u, y, z\}$ of $C_\nu$ is also a face of $C'_\nu = C_\nu - e$. The drawing of $C'_\nu$ was obtained via the transversal pair of bipolar orientations of $C'_\nu$, with red/blue edges corresponding to horizontal/vertical edges. It is well-known that any inner face in a transversal pair of bipolar orientations has at least one red and at least one blue edge. So $\{u, y, z\}$ is drawn with at least one horizontal and at least one vertical edge, say two edges are vertical and one is horizontal (the other case is similar). Now consider which vertex of $\{u, y, z\}$ is the one where both incident edges of triangle $\{u, y, z\}$ are vertical.

If vertex $u$ has two incident vertical edges $(u, y)$ and $(u, z)$, then $(y, z)$ is drawn horizontally. We can then find a feasible region within the thick line of sight of edge $(y, z)$. See Fig 6c.

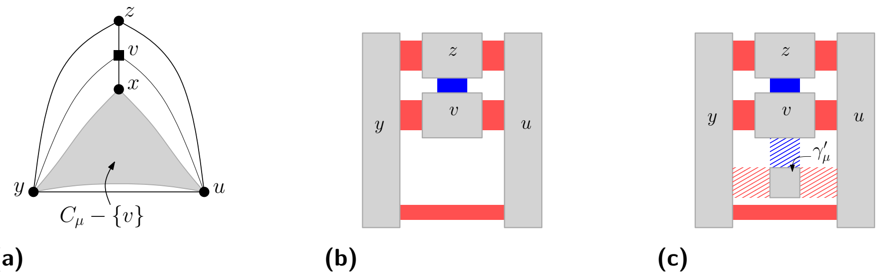


**Figure 7** Illustration for the proof of Observation 12.

Now assume that $y$ has two incident vertical edges $(y, u)$ and $(y, z)$ (the case of vertex $z$ is symmetric). For ease of description, assume that $y$ is above $u$ and $z$, and $u$ is to the left of $z$. If the top side of $u$ is strictly above the top side of $z$, then it is easy to find a feasible region within the thick line of sight of edge $(y, z)$ into which we can merge $\gamma'_\nu$ after rotating it; see Fig. 6d. If the top side of $u$ is below (or at the same height) of the top side of $z$, then we modify the drawing to create a suitable region using a zig-zag-slide. The dividing curve $D$ is defined as follows. Insert a vertical edge just left of $z$, ranging from just above $(u, z)$ (hence below the top of $u$ since we have thick lines of visibility) to just above the top of $z$. Expand it with a horizontal ray leftward from the lower end and a horizontal ray rightward from the upper end. By sliding upward sufficiently far, we hence achieve that the top of $u$ is above the one of $z$, which allows us to find a feasible region as previously. See Fig 6e. ◀

By Lemma 11, for every child $\mu$ of $\nu$ we can find a feasible region into which we merge (after suitable scaling) the drawing $\gamma_\mu - \{u, y, z\}$. This gives the RVR of $C_\nu \cup G_\mu$. It remains to prove the 4-sides-condition for an inner vertex $v$ of $C_\nu \cup G_\mu$. This holds clearly if $v$ was an inner vertex of either $C_\nu$ or $G_\mu$. If it is neither, then one can argue that $v$ is involved in two separating triangles and with our choice of surround-edge the merging ensures the 4-sides-condition. See Fig. 7 and [2] for details. So we have:

▶ **Observation 12.** *In the RVR of $C_\nu \cup G_\mu$, the 4-sides-condition holds for every dummy-vertex that is an inner vertex of $C_\nu \cup G_\mu$.*

### 3.4.6 Undoing the planarization

Once all merging is completed, the drawing $\Gamma_\rho$ at the root gives an RVR of $G^+$. To turn this into an RVR of $G$, we must undo the planarization, hence remove the dummy-vertices and replace them with a crossing. Since dummy-vertices are surrounded by a kite, none of them is on the outer face of $G^+$. So any dummy-vertex $z$ is an inner vertex of $G^+$ and its rectangle $R_z$ satisfies the 4-sides-condition. Let $e_W, e_N, e_E$ and $e_S$ be the four edges incident at the west/north/east and south side of $R_z$. If $e_W$ and $e_E$ have the same $y$-coordinate, and $e_N$ and $e_S$ have the same $x$-coordinate, then we can simply remove $R_z$, extend the edges, and obtain the desired crossing.

So assume that $e_W$ and $e_E$ have different $y$-coordinates, with (say) the $y$-coordinate $y_W$ of $e_W$ larger than the $y$-coordinate $y_E$ of $e_E$. Construct a dividing curve $D$ by starting with a vertical line segment inside $R_z$, ranging from $y$-coordinate $y_W + \varepsilon$ to $y$-coordinate $y_E - \varepsilon$, where $\varepsilon > 0$ is chosen small enough that the segment is inside $R_z$. Attach a leftward horizontal ray at the top and a rightward horizontal ray at the bottom. Now apply a zig-zag-slide of length $y_W - y_E$. This moves $e_E$ upward while keeping $e_W$ stationary, and hence aligns the two edges. See also Fig. 5, where this is illustrated for $e_W = e$ and $e_E = e'$. With another zig-zag-slide (with the dividing curve using a horizontal line segment and vertical rays), we similarly can align $e_N$ and $e_S$, if needed. After this, remove the dummy-vertex to obtain the crossing. Repeating this at all dummy-vertices, and finally deleting all added edges, gives the RVR of $G$, as no crossings are created except where dummy-vertices were removed. This ends the description of algorithm `1PRVDrawer`. We remark that with minor changes `1PRVDrawer` can also handle general (not 3-connected) 1-plane graphs, see [2].

▶ **Lemma 13.** *Let $G$ be a 1-plane graph $G$ and with no B-/W-/T-configuration as a subgraph. Then algorithm `1PRVDrawer` computes an RVR of $G$.*

Lemma 13 shows the sufficieny of Theorem 4.

## 3.5 Area, time complexity and testing

Algorithm `1PRVDrawer` does not obviously have linear run-time, since we repeatedly change the entire drawing, especially when applying zig-zag-slides. Also, coordinates may get very small when merging, resulting (after re-scaling to be integral) in a very large area. However, both of these become non-issues if we store RVRs implicitly, using the orthogonal representations of Observation 2. Using this approach, no zig-zag-slide needs to be performed, since these only change edge lengths (but not angles or adjacencies) and hence give rise to exactly the same orthogonal representation. Therefore, all that is required to merge a subgraph at a separating triangle $\{u, y, z\}$ is to find, for each of $w \in \{u, y, z\}$, the appropriate segment along the boundary of the rectangle of $w$ in $C_\mu$, and to merge here the list of adjacencies and angles that $w$ has in the PSLG of $C_\nu$. This takes constant time for $w$, hence constant time for merging at one separating triangle, and hence linear time overall. All other parts of the algorithm (such as finding separating triangles, computing 4-connected components, finding a transversal pair of bipolar orientations, and finding the rectangular dual representation) take linear time as well. The final conversion of the PSLG into an RVR also takes linear time, and gives linear integral coordinates. Here "linear" measures the size of the PSLG, which in our case is proportional to the number of vertices, edges, and crossings. For 1-planar graphs this is $O(n)$ since 1-planar graphs have at most $4n - 8$ edges [18]. Thus the final RVR has linear coordinates and the area is $O(n^2)$. We conclude:

▶ **Lemma 14.** *Let $G$ be a 1-plane graph $G$ with $n$ vertices and with no B-/W-/T-configuration as a subgraph. Then there exists an $O(n)$-time algorithm that computes an RVR $\Gamma$ of $G$ that has $O(n^2)$ area.*

We finally sketch how to test whether a given 1-plane graph $G$ contains any B-configuration, any W-configuration, or any T-configuration. Hong *et al.* [13] show an algorithm to detect and report every possible B-configuration and W-configuration in $O(n)$ time, where $n$ is the number of vertices of $G$. Thus, in what follows we describe how to detect and report a T-configuration, assuming that $G$ contains no B-configuration and no W-configuration. Let $G^+$ be the plane triangulated graph obtained from $G$ by applying the planarization and triangulation step described in Section 3.4. Recall that $G^+$ contains a T-configuration if and only if $G$ does. Suppose $G^+$ contains a T-configuration $t$ with outer vertices $a, b, c$ and inner vertices $d, e, f, g, h, i$, as in Fig. 2d. Since we added a kite around each crossing, vertices $a, b, c$ induce a triangle in $G^+$. Thus to detect a T-configuration in $G^+$, we first list all triangles of $G^+$. The triangle is a T-configuration if and only the three faces on the inside have dummy-vertices as their third vertex, which can be tested in $O(1)$ time per triangle. The only difficulty is hence to list all triangles efficiently. If $G$ is 3-connected, then $G^+$ is simple and this can be done in $O(n)$ time (see e.g. [6]). The case when $G$ is not 3-connected is described in [2]. This, together with Lemma 14, implies Theorem 5.

## 4 Variable Embedding Setting

Alam *et al.* [1] proved that if $G$ is 3-connected, then it is possible to reroute its edges so as to remove all B-configurations and W-configurations but at most one and hence obtain a straight-line 1-planar drawing. We can show that this is not true for visibility representations. All the proofs for this section can be found in [2].

▶ **Theorem 15.** *Let $k \geq 1$ be an integer. There exists a 3-connected 1-planar graph $G_k$ with $n = 6k$ vertices with at least $k - 1$ T-configurations in every possible 1-planar embedding.*

The following stronger corollary can also be proved.

▶ **Corollary 16.** *Let $k \geq 1$ be an integer value. There exists a 3-connected 1-planar graph $G_k$ with $n = 6k$ vertices that does not admit any 1-planar embedding that can be represented as an RVR unless we delete at least $k = n/6$ edges.*

On the positive side, we can mirror the results of Alam *et al.* for visibility representations by increasing the connectivity one more.

▶ **Theorem 17.** *Every 4-connected 1-planar graph $G$ admits a 1-planar embedding that can be represented as an RVR, except for at most one edge.*

▶ **Corollary 18.** *Every optimal 1-plane graph (i.e., a 1-plane graph with $4n - 8$ edges) admits an RVR, except for one edge.*

## 5 Conclusions and Open Problems

In this paper, we studied rectangle visibility representations of non-planar graphs with a fixed embedding. We showed that we can test in polynomial time whether a graph has such a representation. Of special interest to us were 1-planar graphs; here we can give a linear-time algorithm to test the existence of visibility representations if the embedding is fixed. Moreover, in case of a negative answer the algorithm provides a witness in form of either a B-configuration, W-configuration, or T-configuration. We also briefly studied 1-planar graphs without fixed embeddings.

The most pressing open problem is whether we can restrict the drawings less and still test for the existence of visibility representations? Most importantly, if we fix the rotational scheme and outer face, but *not* order in which crossing occur (and perhaps not even which edges cross), is it possible to test whether a visibility representation respecting the rotational scheme and outer face exists? The NP-hardness proof of Shermer [20] does not hold if the rotational scheme is fixed, since it uses a reduction from linear-arboricity-2, and fixing the rotational scheme would severely restrict the possible ways of splitting a graph into two linear forests. The orthogonal representation approach utterly fails if the order of crossings is not fixed, since it requires planarization as a first step.

Secondly, can we characterize for more graph classes exactly when they have a rectangle visibility representation? Notice that Lemma 6 does not use 1-planarity, and hence gives a necessary condition for any subgraph of a graph $G$ (with a fixed embedding) to have an RVR. A second necessary condition stems from that as soon as edges may have 2 or more crossings, the edge-parts between the crossings may be non-trivial and have cycles; clearly the graph formed by these crossings must be bipartite if $G$ has a visibility representation. Are these two conditions sufficient, and if not, can we find necessary and sufficient conditions, at least for some restricted graph classes such as 2-planar and fan-planar graphs?

## References

1 Md. Jawaherul Alam, Franz J. Brandenburg, and Stephen G. Kobourov. Straight-line grid drawings of 3-connected 1-planar graphs. In *GD 2013*, volume 8242 of *LNCS*, pages 83–94. Springer, 2013.

2 Therese Biedl, Giuseppe Liotta, and Fabrizio Montecchiani. On visibility representations of non-planar graphs. *CoRR*, abs/1511.08592, 2015. URL: `http://arxiv.org/abs/1511.08592`.

3 Therese Biedl, Anna Lubiw, Mark Petrick, and Michael J. Spriggs. Morphing orthogonal planar graph drawings. *ACM Trans. Algorithms*, 9(4):29, 2013. `doi:10.1145/2500118`.

4 Prosenjit Bose, Alice M. Dean, Joan P. Hutchinson, and Thomas C. Shermer. On rectangle visibility graphs. In *GD 1996*, volume 1190 of *LNCS*, pages 25–44. Springer, 1997.

5 Franz J. Brandenburg. 1-Visibility representations of 1-planar graphs. *J. Graph Algorithms Appl.*, 18(3):421–438, 2014.

6 Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14(1):210–223, 1985.

7 Sabine Cornelsen and Andreas Karrenbauer. Accelerated bend minimization. *J. Graph Algorithms Appl.*, 16(3):635–650, 2012.

8 Alice M. Dean, William S. Evans, Ellen Gethner, Joshua D. Laison, Mohammad Ali Safari, and William T. Trotter. Bar $k$-visibility graphs. *J. Graph Algorithms Appl.*, 11(1):45–59, 2007.

9 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice-Hall, 1999.

10 Pierre Duchet, Yahya Ould Hamidoune, Michel Las Vergnas, and Henry Meyniel. Representing a planar graph by vertical lines joining different levels. *Discrete Math.*, 46(3):319–321, 1983.

11 William S. Evans, Michael Kaufmann, William Lenhart, Tamara Mchedlidze, and Stephen K. Wismath. Bar 1-visibility graphs vs. other nearly planar graphs. *J. Graph Algorithms Appl.*, 18(5):721–739, 2014.

12 Éric Fusy. Transversal structures on triangulations: A combinatorial study and straight-line drawings. *Discrete Math.*, 309(7):1870–1894, 2009.

13 Seok-Hee Hong, Peter Eades, Giuseppe Liotta, and Sheung-Hung Poon. Fáry's theorem for 1-planar graphs. In *COCOON 2012*, volume 7434 of *LNCS*, pages 335–346. Springer, 2012.

14 Goos Kant. A more compact visibility representation. *Int. J. Comput. Geometry Appl.*, 7(3):197–210, 1997.

15 Goos Kant and Xin He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theor. Comput. Sci.*, 172(1-2):175–193, 1997.

16 Goos Kant, Giuseppe Liotta, Roberto Tamassia, and Ioannis G. Tollis. Area requirement of visibility representations of trees. *Inf. Process. Lett.*, 62(2):81–88, 1997.

17 Ralph H. J. M. Otten and J. G. Van Wijk. Graph representations in interactive layout design. In *IEEE ISCSS*, pages 914–918. IEEE, 1978.

18 János Pach and Gáza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997.

19 Pierre Rosenstiehl and Robert Endre Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discr. & Comput. Geom.*, 1:343–353, 1986.

20 Thomas C. Shermer. On rectangle visibility graphs. III. External visibility and complexity. In *CCCG 1996*, pages 234–239. Carleton University Press, 1996.

21 Ileana Streinu and Sue Whitesides. Rectangle visibility graphs: Characterization, construction, and compaction. In *STACS 2003*, volume 2607 of *LNCS*, pages 26–37. Springer, 2003.

22 Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Computing*, 16(3):421–444, 1987.

23 Roberto Tamassia and Ioannis G. Tollis. A unified approach to visibility representations of planar graphs. *Discr. & Comput. Geom.*, 1(1):321–341, 1986.

24 Carsten Thomassen. Plane representations of graphs. In *Progress in Graph Theory*, pages 43–69. AP, 1984.

25 Carsten Thomassen. Rectilinear drawings of graphs. *J. Graph Theory*, 12(3):335–341, 1988.

26 Stephen K. Wismath. Characterizing bar line-of-sight graphs. In *SoCG 1985*, pages 147–152. ACM, 1985.