

Visibility Representations of Toroidal and Klein-bottle Graphs

Therese Biedl^[0000-0002-9003-3783] *

David R. Cheriton School of Computer Science, University of Waterloo, Waterloo,
Ontario N2L 3G1, Canada. biedl@uwaterloo.ca

Abstract In this paper, we study visibility representations of graphs that are embedded on a torus or a Klein bottle. Mohar and Rosenstiehl showed that any toroidal graph has a visibility representation on a flat torus bounded by a parallelogram, but left open the question whether one can assume a rectangular flat torus, i.e., a flat torus bounded by a rectangle. Independently the same question was asked by Tamassia and Tollis. We answer this question in the positive. With the same technique, we can also show that any graph embedded on a Klein bottle has a visibility representation on the rectangular flat Klein bottle.

1 Introduction

Visibility representations are one of the oldest topics studied in graph drawing. Introduced as *horvert-drawings* by Otten and Van Wijk in 1978 [21], and independently as *S-representations* by Duchet, Hamidoune, Las Vergnas and Meyniel in 1983 [10], they consist of assigning disjoint horizontal segments to vertices and disjoint vertical segments to every edge such that for each edge the segment ends at the two vertex-segments of its endpoints and intersects no other vertex-segment. (Fig. 2(d) gives an example.) Later papers studied exactly which planar graphs have such visibility representations [23,24,27] and generalized them to the rolling cylinder [26], Möbius band [7], projective plane [16] or torus [20]. (There are numerous other generalizations, e.g. to higher dimensions [3], or permitting rectangles for vertices and horizontal and vertical edges [4], or permitting edges to go through a limited set of vertex-segments [8].)

The motivation for the current paper is the work by Mohar and Rosenstiehl [20], who showed that any *toroidal graph* (i.e., a graph that can be drawn on a torus without crossings) has a visibility representation on the *flat torus*, i.e., a parallelogram Q where opposite edges have been identified. They explicitly stated as open problem whether the same holds for a *rectangular flat torus*, i.e., where Q must be a rectangle—their method cannot be generalized to this case. (See also Fig. 5.) The same question was asked independently earlier by Tamassia and Tollis [26]. This paper answers this question in the positive.

Theorem 1. *Let G be a toroidal graph without loops. Then G has a visibility representation on the rectangular flat torus.*

* Supported by NSERC. The author would like to thank Sam Barr for helpful input.

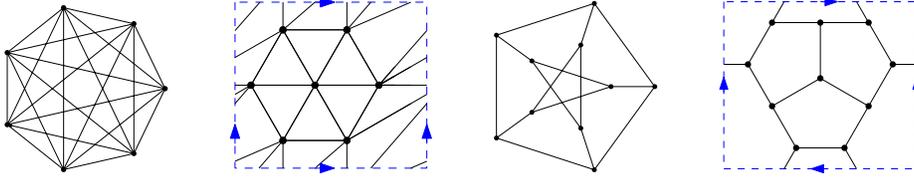


Figure 1. The complete graph K_7 embedded on the rectangular flat torus and the Petersen-graph embedded on the rectangular flat Klein bottle.

There are quite a few graph drawing results for toroidal graphs; see Castelli Aleardi et al. [5] and the references therein for increasingly better results for straight-line drawings. Their approach is to convert the toroidal graph into a planar graph by deleting edges, then draw this planar graph, and then reinsert the edges. (Other papers [16,20] instead use a reduction approach, where the graph-size is reduced while staying in the same graph class until some small graph is reached, draw this graph, and then undo the reduction in the drawing.) We follow the first approach (i.e., delete edges to make the graph planar), but face a major challenge when wanting to reinsert an edge (v, w) . For this, we need the segments of v and w to be visible across the horizontal boundary of the fundamental rectangle, and in particular, to share an x -coordinate. We achieve this by keeping two halves of each removed edge, connecting corresponding half-edges along paths, and then forcing these paths to be drawn along columns; the ability to do so may be of independent interest.

2 Background

We assume familiarity with graph theory and planar graphs, see for example Diestel’s book [9]. Throughout, let $G = (V, E)$ be a connected graph without loops, with $|V| = n$ and $|E| = m$. A *map* M on a surface Σ is a 2-connected graph G together with an embedding of G in Σ such that every *face* (i.e., connected region of $\Sigma \setminus M$) is bounded by a simple cycle. Maps correspond naturally to rotation systems on the underlying graphs, up to homomorphisms among the embeddings [15]. Here a *rotation system* is a set of cyclic permutations ρ_v (for $v \in V$) where ρ_v corresponds to the clockwise cyclic order in which the edges incident to v emanate from v in the embedding. For ease of description we often assume that we have a map, though all algorithmic steps could be performed on the rotation system alone.

We study surfaces that have a *flat representation* consisting of a *fundamental parallelogram* Q in the plane with some sides identified. (We may assume that two sides of Q are horizontal, hence Q has a left/right/top/bottom side.) A (*standing*) *flat cylinder* is obtained by identifying the left and right side of Q in the same direction (bottom-to-top). (We usually omit ‘standing’ since we will not discuss other kinds.) A *flat torus* is obtained from a flat cylinder by identifying the top and bottom side in the same direction (left-to-right), while a

flat Klein bottle is obtained from a flat cylinder by identifying the top and bottom side in opposite direction. Figs. 1, 5, 6 give some examples. A *rectangular flat torus* [*rectangular Klein bottle*] is a flat torus [flat Klein bottle] for which the fundamental parallelogram Q is required to be a rectangle.

Flat representations carry the local geometry of the plane; in particular when we speak of a *segment* or an *x-interval* then we specifically permit it to go across a side of the fundamental parallelogram Q . So for example in a flat cylinder $Q = [0, w] \times [0, h]$, an *x-interval* can have the form $[x', x'']$ for two *x*-coordinates $x' < x''$, but it can also have the form $[0, x''] \cup [x', w]$ for some $x'' < x'$. A *row/column* of Q is a horizontal/vertical line with integer coordinate that intersects the interior of Q .

A *visibility representation* of a graph G is a mapping of vertices into non-overlapping horizontal segments (called *vertex-segments*) and of edges of G into non-overlapping vertical segments (called *edge-segments*) such that for each edge (u, v) , the associated edge-segment has its endpoints on the vertex-segments corresponding to u and v and it does not intersect any other vertex-segment.

3 Creating Visibility Representations

We first give an outline of our approach. Quite similar to what was done for straight-line drawings of toroidal graphs [5], we remove a set of edges to convert the given graph into a planar graph. In contrast to the earlier work, we keep the edges but split each of them into two ‘half-edges’ that end at two new vertices s, t (Section 3.1). We will later need to re-connect these half-edges, and to this end, choose a ‘path-system’ that connects each pair of half-edges while keeping all the paths non-crossing and (after duplicating some edges) edge-disjoint (Section 3.2). Then we create a visibility representation on the flat cylinder for which these paths are drawn vertically. To be able to do so we first must argue that we can find an *st*-order that enumerates vertices of all paths in order (Section 3.3). Then we build the visibility representation (Section 3.4). Removing the segments of s and t and possibly inserting more columns gives the desired visibility representation. Figs. 2 and 3-4 illustrate the approach for K_7 and the Petersen-graph.

3.1 Making the Graph Planar

In this section we explain how to modify the input graph G to make it planar. We assume that G has no loop and comes embedded on a flat realization Q (either a torus or a Klein bottle). We first modify this embedding to achieve the following: (1) Every face is bounded by a simple cycle, so the embedding is a map. (2) No edge crosses the horizontal boundary of Q twice. (3) Parallelogram Q is a rectangle. (4) No vertex lies on the boundary of Q . (5) Edges intersect the boundary of Q in a finite set of points, and do not use a corner of Q . Conditions (1-5) can easily be achieved if arbitrary curves are allowed for edges as follows: (1) holds after adding sufficiently many edges (which can be deleted in the

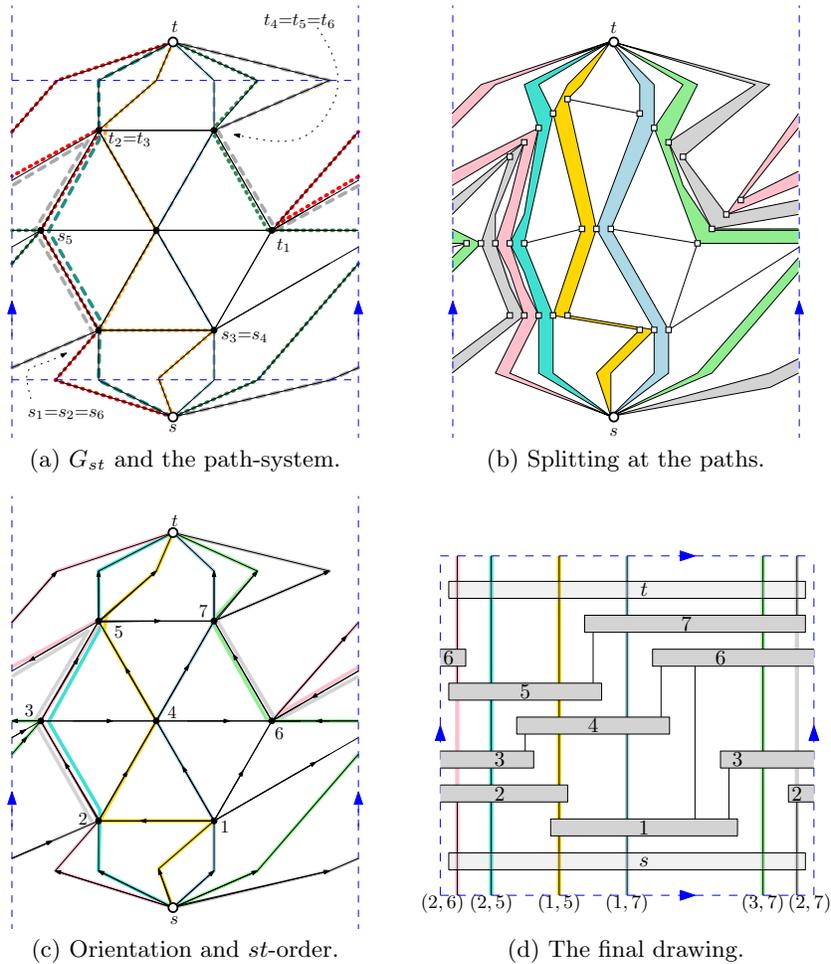


Figure 2. The construction for the complete graph K_7 .

final visibility representation), (2) can be achieved by re-routing the horizontal boundary of Q along a so-called *tambourine* [5], (3) holds after a shear and (4-5) hold after locally re-routing.

Assume first that G is toroidal, so Q is a rectangular flat torus. Enumerate the edges that intersect the bottom side of Q as (s_i, t_i) (for $i = 1, \dots, d$) from left to right, named such that part of the edge that goes upward from the bottom side ends at s_i for $i = 1, \dots, d$. (This is feasible by condition (2) above.) Create a new graph G_{st} by removing edges (s_i, t_i) for $i = 1, \dots, s$, adding a new vertex t incident to t_1, \dots, t_d and a new vertex s incident to s_1, \dots, s_d . See Fig. 2(a).

Now assume that G is embedded on a rectangular flat Klein bottle Q instead. We construct G_{st} in almost the same way, but the enumeration of edges is different. Let the edges that cross the bottom side of Q be $(s_1, t_d), \dots, (s_d, t_1)$

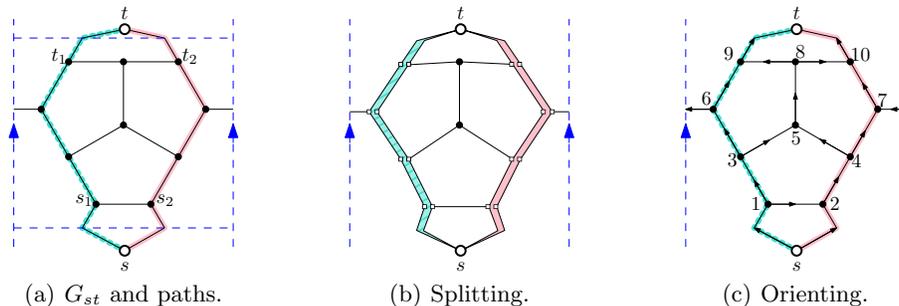


Figure 3. The first few steps for the Petersen-graph from Fig. 1(b).

from left to right, named such that the part of the edge that goes upward from the bottom side ends at s_i for $i = 1, \dots, d$. Since the top and bottom sides of Q are identified in opposite direction, the order of edges along the top side of Q is $(s_d, t_1), \dots, (s_1, t_d)$ from left to right. Remove these edges and replace them by a vertex s incident to s_1, \dots, s_d and a vertex t incident to t_1, \dots, t_d . See Fig. 3(a).

In both cases, by placing t above the top side of Q and s below the bottom side of Q , we obtain an embedding of G_{st} on the flat cylinder, so it is a *plane* graph (i.e., drawn on the plane with a fixed embedding). The edges incident to s lead to s_1, \dots, s_d (in clockwise order) and the edges incident to t lead to t_1, \dots, t_d (in counter-clockwise order).

Observation 1. *Graph G_{st} is 2-connected.*

Proof. Since G_{st} is a plane graph, 2-connectivity is equivalent to all faces being bounded by a simple cycle. This holds for all faces of G by assumption. The only faces of G_{st} that are not in G are those incident to s and t . These consist of part of the boundary of a face of G , plus two newly added edges that both end at s (or both end at t). So the boundary of these faces are simple cycles as well. \square

3.2 Choosing Paths

We now show how to choose a set Π of paths in G_{st} that satisfy some properties. A path is called *simple* if no vertex repeats. Two simple edge-disjoint paths π, π' are *non-crossing* if at any vertex v that is interior to both the paths only touch, i.e., the edges of the paths appear in order π, π, π', π' in ρ_v .

Lemma 1. *There exists a planar graph \hat{G} (obtained by duplicating edges of G_{st}) and a set of simple edge-disjoint non-crossing paths π_1, \dots, π_d in \hat{G} such that path π_i begins with (s, s_i) and ends with (t_i, t) for $i = 1, \dots, d$.*

Before giving the proof, we need to define the operation of *splitting* a map at a path π (also used in Figs. 2(b) and 3(b)). Temporarily direct π from one end

to the other. Duplicate all interior vertices of π (say vertex v becomes v^ℓ and v^r) and duplicate all edges of π correspondingly. For any interior vertex v of π , and any edge e incident to v but not on π , we re-connect e to end at v^ℓ [v^r] if e occurs before [after] the outgoing edge of π at v when enumerating ρ_v beginning with the incoming edge of π on v . Splitting at π creates a new face f_π bounded by the two copies of π .

Proof. Let π be a simple path that begins with (s, s_1) and ends with (t_1, t) ; this exists since G_{st} is 2-connected. Temporarily split graph G_{st} at π to obtain a planar graph \tilde{G} . The resulting new face f_π contains both s and t ; for ease of description we assume that f_π is the outer-face of \tilde{G} .

Let \tilde{G}^+ be the graph obtained from \tilde{G} by replacing any edge e that is not incident to s or t by a multi-edge that has $d+1$ copies of e . Any s - t -cut of \tilde{G}^+ either consists of the edges incident to s (then it has size $d+1$ since (s, s_1) exists twice in \tilde{G}) or of the edges incident to t (then it likewise has size $d+1$), or it contains some edge e not incident to either s or t and so has size at least $d+1$. By the max-flow-min-cut theorem therefore \tilde{G}^+ has a flow of value $d+1$ from s to t ; equivalently, it has $d+1$ edge-disjoint paths π_1, \dots, π_{d+1} from s to t . Since s and t are both on the outer-face we can find these paths using *right-first search* [22]; this will automatically make them crossing-free.

Since the paths are crossing-free and use all edges incident to s, t , and since s and t are on the outer-face, there is no choice which pair of edges must be the first and last on each path. The clockwise order of edges at s (beginning after the outer-face) is $(s, s_1^r), \dots, (s, s_d), (s, s_1^\ell)$. The counter-clockwise order of edges at t (beginning after the outer-face) is $(t, t_1^r), \dots, (t, t_d), (t, t_1^\ell)$. Therefore path π_i begins with (s, s_i) and end with (t_i, t) for $i = 2, \dots, d$, while π_1 and π_{d+1} use the copies of s_1 and t_1 .

To obtain \hat{G} , re-combine any two vertices v^ℓ and v^r that resulted from splitting an interior vertex v of π , and keep all edges of \tilde{G}^+ except (s, s_1^ℓ) and (t_1^ℓ, t) . Since these two edges were used by π_{d+1} , they were used by no other path in π_1, \dots, π_d , and we have hence obtained our desired path-system. \square

3.3 A Path-Constrained st -order

By Lemma 1, we can fix a supergraph \hat{G} of G_{st} and a *path-system* Π , i.e., a set of simple edge-disjoint non-crossing paths from s to t . To draw \hat{G} , we add vertices one-by-one, and to draw the paths in Π vertically, we require a vertex-order with special properties.

We need some definitions. A *bipolar orientation* is an assignment of directions to the edges that is acyclic and has exactly one source and one sink. An *st -order* is a vertex order v_1, \dots, v_n such that orienting all edges from the lower-indexed to the higher-indexed vertex gives a bipolar orientation. Vice versa, for any bipolar orientation, enumerating the vertices in topological order gives an st -order. It is well-known that any 2-connected graph has a bipolar orientation, even if we fix a-priori which vertices should be the source and sink [17]; it can be found in linear time [11].

We say that a bipolar orientation *respects a path system* Π if every path in Π is directed from s to t in the bipolar orientation. We phrase the following result for an arbitrary graph H since it does not depend on the graph stemming from a toroidal or Klein-bottle graph and may be of independent interest.

Lemma 2. *Let H be a 2-connected plane graph with two vertices $s \neq t$. Let Π be a set of simple edge-disjoint crossing-free paths from s to t . Then H has a bipolar orientation that respects Π and has source s and sink t .*

Proof. Consider the graph \hat{H} obtained from H by splitting H at each path in Π . See Figs. 2(b) and 3(b). Any face of \hat{H} is either a face of H (then it is a simple cycle since H is 2-connected) or f is bounded by the two copies of some path $\pi \in \Pi$ (then it is a simple cycle since π is simple). So \hat{H} is 2-connected and has a bipolar orientation \hat{D} with source s and sink t .

It is well-known [24] that in \hat{D} any face has a unique source and sink. In any face f_π bounded by two copies of some $\pi \in \Pi$, the unique source is s and the unique sink is t . Therefore both copies of π are directed from s to t and undoing the splitting gives the desired orientation. \square

3.4 Path-Constrained Visibility Representations

In this section, we give an easy construction of a visibility representation on the flat cylinder where a given path-system Π is drawn vertically. Formally, we say that a path π lies *on an exclusive column* ℓ (in a visibility representation Γ) if all edges of π are represented by segments on ℓ , and column ℓ intersects no vertex- or edge-segment except the ones that belong to vertices/edges of π .

Our approach to create visibility representations is quite different from prior constructions [16,20,21,23,24,26,27], which either read the coordinates for the segments directly from the orientation (using the length of the longest paths in the primal and dual graph), or reduced the graph (or its dual) by removing an edge somewhere in the graph, creating a representation recursively, and expanding. In contrast to this, we use here an incremental approach which resembles more the incremental approaches taken for straight-line drawings [5,13] or orthogonal drawings [2]. This uses a vertex ordering and adds the vertices to the drawing one-by-one.

Theorem 2. *Let H be a 2-connected plane graph with two vertices s, t and let Π be a set of simple edge-disjoint non-crossing paths from s to t . Then H has a visibility representation on the flat cylinder such that each $\pi \in \Pi$ lies on an exclusive column.*

Proof. Fix a bipolar orientation using Lemma 2 and extract an st -order v_1, \dots, v_n from it; we know $v_1 = s$ and $v_n = t$ and the numbers along any path in Π increase from s to t . For $i = 1, \dots, n$ let H_i be the subgraph induced by v_1, \dots, v_i and let the *cut* $E_{i:i+1}$ be the set of all edges (v_h, v_j) with $h \leq i < j$. There is a natural cyclic order of the edges in $E_{i:i+1}$ implied by the embedding of H (specifically, if we contracted the vertices v_1, \dots, v_i into a supernode, then the

order of $E_{i:i+1}$ would be the clockwise order of edges at this supernode). We will use induction on i to create a visibility representation of H_i on a flat cylinder that satisfies the following for $i < n$:

1. Every edge $e = (v_h, v_j)$, $h < j$ in cut $E_{i:i+1}$ is associated with a column that intersects v_h and that is empty above v_h .
2. The left-to-right order of columns associated with $E_{i:i+1}$ respects the cyclic order of edges in $E_{i:i+1}$.
3. For any path $\pi \in \Pi$, the sub-path of π in H_i lies on an exclusive column, and the same column is associated with the unique edge of π in $E_{i:i+1}$.

Fig. 4(a-b) illustrates the following construction. For $i = 1$, we create the desired visibility representation simply by defining a horizontal line segment $s(v_1)$ for v_1 with y -coordinate 0 and width $|E_{1:2}|$, and assigning columns intersecting $s(v_1)$ to edges in $E_{1:2}$ in the correct order.

For $i > 1$, assume we have created a visibility representation of H_{i-1} already. Define edge-sets $E_i^- := \{(v_h, v_i) : h < i\}$ and $E_i^+ := \{(v_i, v_j) : i < j\}$; the former is non-empty by $i > 1$ since we have an st -order. It is well-known [17] that E_i^- is consecutive in the cyclic order of edges in $E_{i-1:i}$. By the invariant therefore there exists an x -interval X_i on the flat cylinder that intersects all columns associated with edges in E_i^- in its interior and intersects no other columns associated with $E_{i-1:i}$. Define the segment $s(v_i)$ of v_i to have x -range X_i and a y -coordinate that is higher than the one of all its neighbours in E_i^- . These edges can then be completed along their associated columns.

To associate columns with E_i^+ , we insert new columns as needed. First consider any edge $e \in E_i^+$ in some path $\pi \in \Pi$. Since π begins at s and $i > 1$, and since indices increase along π , some edge $e' \in E_i^-$ also belongs to π . Associate the column of e' with e . Notice that this associates columns in the correct order, because if multiple paths $\pi_1, \dots, \pi_k \in \Pi$ all went through v_i , then the counter-clockwise order of their edges in E_i^- at v_i must be the same as the clockwise order of their edges in E_i^+ at v_i , otherwise two of these paths would cross at v_i .

Now consider any edge $e \in E_i^+$ that does not belong to a path in Π . Assign a ray upward from $s(v_i)$ to e , choosing rays such that all edges in E_i^+ use distinct rays/columns and their order reflects the order of edges at v_i . By stretching horizontal segments as needed, we can re-assign coordinates so that all inserted rays lie on integer coordinates, hence become new columns. This gives the desired visibility representation of H_i . \square

3.5 Putting It All Together

We now have all ingredients to prove our main result (Theorem 1): Any toroidal graph G without loops has a visibility representation on the rectangular flat torus. See Fig. 2 for the entire process.

Proof. Add edges to G until all its faces are simple cycles. As described in Subsections 3.1-3.4, split G at edges (s_i, t_i) (for $i = 1, \dots, d$) to obtain G_{st} ,

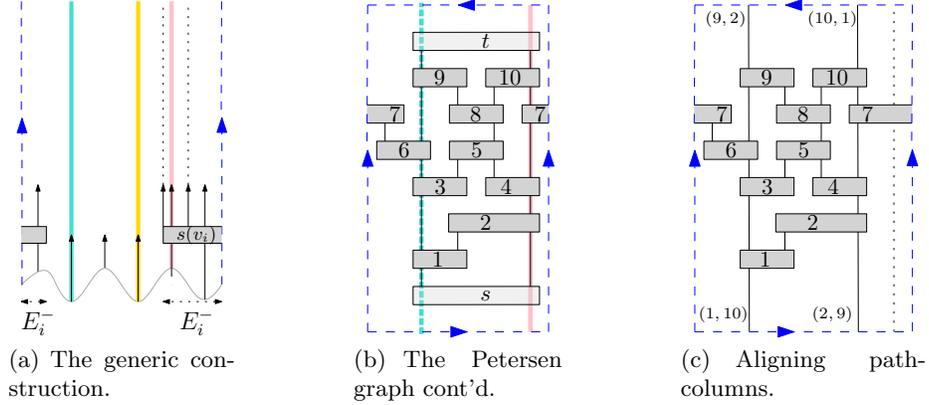


Figure 4. Creating visibility representations.

find a supergraph \hat{G} with a path-system Π where path π_i begins with (s, s_i) and ends with (t_i, t) , find an orientation that respects Π , and find a visibility representation Γ of \hat{G} on the flat cylinder Q such that π_i is drawn along an exclusive column ℓ_i . Remove the segments that represent s and t and complete (s_i, t_i) along column ℓ_i . After re-interpreting Q as a rectangular flat torus this gives the desired visibility representation of G after deleting all added edges. \square

With a bit more care when reconnecting edges, the same approach also works for Klein-bottle graphs.

Theorem 3. *Let G be a graph without loops embedded on the Klein bottle. Then G has a visibility representation on the rectangular flat Klein bottle.*

Proof. Exactly as in the previous proof, create a visibility representation Γ of \hat{G} on the flat cylinder Q such that π_i is drawn along an exclusive column ℓ_i . Remove the segments that represent s and t and extend (s_i, s) and (t_i, t) along ℓ_i until they reach the horizontal boundary of Q .

We are not quite done yet, because we must ensure that column ℓ_i ‘lines up’ with column ℓ_{d+1-i} (for $i = 1, \dots, \lfloor d/2 \rfloor$) so that edges (s_i, t_{d+1-i}) and (s_{d+1-i}, t_i) are connected correctly when interpreting Q as the flat Klein bottle. This is easily achieved by inserting columns. Namely, assume Q has x -range $[0, w]$ and let $x(\ell)$ denote the x -coordinate of column ℓ . For $i = 1, \dots, \lfloor d/2 \rfloor$, while $x(\ell_i) < w - x(\ell_{d+1-i})$, insert an empty column to the left of ℓ_i , and while $x(\ell_i) > w - x(\ell_{d+1-i})$, insert an empty column to the right of ℓ_{d+1-i} . See Fig. 4(c). This maintains distances of $\ell_1, \dots, \ell_{i-1}$ to the left boundary and distances of $\ell_{d+2-i}, \dots, \ell_d$ to the right boundary of Q . So performing this for $i = 1, \dots, \lfloor d/2 \rfloor$ gives the desired visibility representation on the flat Klein bottle. \square

We note here that our visibility representations exactly respect the given embedding. Under this restriction, the condition ‘no loops’ cannot be avoided.

(This was essentially observed by Mohar and Rosenstiehl [20] already.) Namely, let M_0 be a graph with a single vertex v and two loops ℓ_1, ℓ_2 such that $\rho_v = \langle \ell_1, \ell_2, \ell_1, \ell_2 \rangle$. This is toroidal, but has no visibility representation on the rectangular flat torus that respects the embedding since the rotation scheme at v in such an embedding is necessarily $\ell_1, \ell_1, \ell_2, \ell_2$.

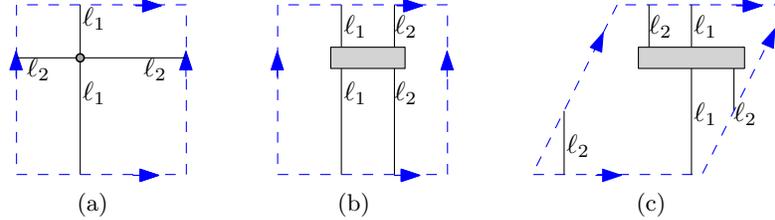


Figure 5. (a) Graph M_0 . (b) The only possible visibility representation on a rectangular flat torus. (c) An embedding-preserving visibility on the flat torus.

3.6 Grid-Size

We can give a bound on the grid-size of Theorem 1, assuming that the input is already a map (i.e., all faces are simple cycles). We say that a visibility representation has *grid-size* $w \times h$ if the fundamental rectangle Q intersects w columns and h rows, not counting the boundaries of Q . In our current approach, the visibility representation Γ_{st} of G_{st} uses significantly more area than it needs to since we may duplicate quite a few edges of G_{st} to obtain the path system (see also the discussion below). However, as for all visibility representations, one should apply compaction steps (similar as for VLSI design [18]) to reduce the size of the drawing. We claim that after doing this, the visibility representation Γ of a toroidal graph G has grid-size at most $(m - n) \times n$.

To see this, observe that we need at most n rows, since assigning row i to vertex v_i will certainly place it high enough and the rows for s, t can be deleted during compaction. As for the number of columns, each column must contain at least one edge, else it could have been deleted. Furthermore, we used a bipolar orientation of \hat{G} , which means that every vertex other than s and t has both an incoming and an outgoing edge. Since \hat{G} is obtained from G_{st} by duplicating edges, the same holds in G_{st} . Vertices s and t are removed in the final visibility representation (but their incident edges remain and are re-combined). With the standard compaction steps, therefore at least one column at each vertex v is used for two edges incident to v . It follows that each vertex saves at least one column, hence the number of columns is $m - n$.

3.7 Run-time

Following the steps of our algorithm, it is very clear that our visibility representations can be found in polynomial time. In fact, the drawing in Theorem 2 can be found in linear time with standard-approaches: do not explicitly maintain the x -coordinates, but store the drawing implicitly by computing x -spans of vertex-segments and x -offsets of edge-segments from the left endpoints of their lower endpoints. The final drawing can then be computed with one pass over the entire graph after all vertices have been placed.

Unfortunately finding the drawings in Theorems 1 and 3 may take superlinear time since the supergraph \hat{G} may have many extra edges. If G_{st} has $\Omega(n)$ disjoint edge-cuts that separate s and t , then each of the $|II|$ paths must duplicate an edge in each edge-cut, leading to $\Omega(m + |II|n)$ edges for \hat{G} . One can show that $|II| \in O(\sqrt{n})$ can be achieved, because any toroidal graph has a non-contractible cycle of length $O(\sqrt{n})$ [1], and we can use such a cycle in the dual graph to find an embedding where $O(\sqrt{n})$ edges cross the horizontal side and hence necessitate a path in II . With this choice we get $|\hat{G}| \in O(n^{1.5})$ and run-time $O(n^{1.5})$.

Reducing this to linear time seems not implausible: we need the paths in II only to steer us towards placing edges in the visibility representation at a suitable place, and it may be possible to encode this in a smaller data structure that permits linear run-time. This remains for future work.

4 Other Drawing Styles

We close the paper by discussing how our results do (or do not) imply results in some other graph drawing styles that are closely related to visibility representations. The first drawing style that we consider are *orthogonal point-drawings*, where vertices are represented by points and every edge is a polygonal curve between its endpoints that uses only horizontal and vertical segments and does not intersect other edges or vertices. (These can only exist if the graph has maximum degree at most four.)

Theorem 4. *Every toroidal graph with maximum degree four has an orthogonal point-drawing on the rectangular flat torus. Every Klein-bottle graph with maximum degree four has an orthogonal point-drawing on the flat Klein bottle.*

Proof. Tamassia and Tollis [25] showed how to create orthogonal point-drawings by starting with a visibility representation and replacing vertex-segments locally by points and polygonal curves that connect to the edge-segments. The exact same transformations can be applied to any visibility representation that lies on a flat representation, so using it with Theorem 1 and Theorem 3 (after subdividing loops, if any) gives the desired orthogonal point-drawings. \square

Two other related drawing styles are grid contact and tessellation representations. A *bipartite graph* has a vertex-partition $V = W \cup B$ such that there are no edges within W or within B . In a *grid contact representation* of a bipartite

graph, the vertices of W and B are assigned to horizontal and vertical segments, respectively, with all segments disjoint except that any segment of one kind may touch at both of its ends an interior point of a segment of the other kind, and such a common point occurs only if the two vertices are adjacent. See Fig. 6(b). It is well-known [12] that every planar bipartite graph has a grid contact representation in the plane, and Mohar and Rosenstiehl [20] showed that any toroidal bipartite graph has a grid contact representation on the flat (not necessarily rectangular) torus. A *tessellation representation* of a graph G is a grid contact representation of the bipartite graph whose vertices are the faces and vertices of G and whose edges are the incidences between them.¹ See Fig. 6(c).

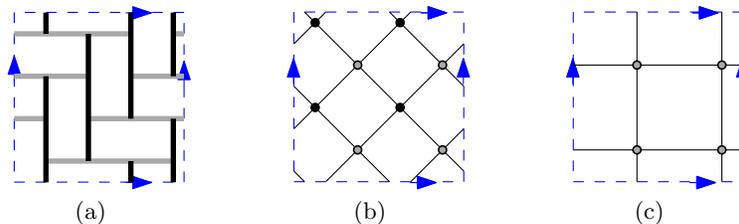


Figure 6. (a) A set of segments that is a grid contact representation of $K_{4,4}$ (shown in (b)) or a tessellation representation of the graph in (c).

Mohar and Rosenstiehl constructed tessellation representations of toroidal graphs (on a flat torus), from which their results on grid contact representations and visibility representations follow easily. They must permit a non-rectangular flat torus because they reduce their graph to M_0 (or another single-vertex graph with loops), which cannot be represented on a rectangular flat torus. But does it help to have no loops?

Conjecture 1. Every toroidal graph without loops has a tessellation representation on the rectangular flat torus.

Conjecture 2. Every bipartite toroidal graph without loops has a grid contact representation on the rectangular flat torus.

At first sight one might think that Theorem 1 implies Conjecture 1, because Mohar and Rosenstiehl [20] show that a visibility representation can be converted to a tessellation representation. Alas, their definition of “visibility representation” uses the ‘strong’ model where *all* visibilities must lead to an edge, hence faces are triangles, and this is vital in their proof. On the positive side, their proof does not affect the shape of the flat representation, so using it one can show that Conjecture 1 holds for toroidal graphs where all faces are triangles.

¹ In contrast to earlier work [20], we use here *weak* models, where not all adjacencies that could be added must exist.

Finally we are interested in *segment intersection representations*, i.e., every vertex is assigned to a segment (of arbitrary slope) on the flat torus, with segments intersecting if and only if the vertices are adjacent. Such representations exist for all planar graphs [6], and one proof of this proceeds by representing a planar graph as the intersection-graph of L-shaped curves in the plane [14] and then converting the L-shaped curves into segments [19]. The corresponding questions on the flat torus appear to be open even if we drop ‘rectangular’:

Question 1. Does every simple toroidal graph have a segment intersection representation on the flat torus?

Question 2. Is every simple toroidal graph the intersection-graph of L-shaped curves on the flat torus?

Question 3. If a graph is the intersection-graph of L-shaped curves on the flat torus, then is it also the intersection-graph of segments on the flat torus?

Finally all these questions could be asked also for graphs embedded on the Klein bottle (or other surfaces, such as the projective plane).

References

1. Albertson, M., Hutchinson, J.: On the independence ratio of a graph. *J. Graph Theory* **2**(1), 1–8 (1978). <https://doi.org/10.1002/jgt.3190020102>
2. Biedl, T., Kant, G.: A better heuristic for orthogonal graph drawings. *Computational Geometry: Theory and Applications* **9**, 159–180 (1998). [https://doi.org/10.1016/S0925-7721\(97\)00026-6](https://doi.org/10.1016/S0925-7721(97)00026-6)
3. Bose, P., Everett, H., Fekete, S., Houle, M., Lubiw, A., Meijer, H., et al.: A visibility representation for graphs in three dimensions. *J. Graph Algorithms Appl.* **2**(3), 1–16 (1998). <https://doi.org/10.7155/jgaa.00006>
4. Bose, P., Dean, A.M., Hutchinson, J.P., Shermer, T.C.: On rectangle visibility graphs. In: North, S.C. (ed.) *Graph Drawing (GD’96)*. Lecture Notes in Computer Science, vol. 1190, pp. 25–44. Springer (1996). https://doi.org/10.1007/3-540-62495-3_35
5. Castelli Aleardi, L., Devillers, O., Fusy, É.: Canonical ordering for graphs on the cylinder, with applications to periodic straight-line drawings on the flat cylinder and torus. *J. Comput. Geom.* **9**(1), 391–429 (2018). <https://doi.org/10.20382/jocg.v9i1a14>
6. Chalopin, J., Gonçalves, D.: Every planar graph is the intersection graph of segments in the plane. In: *ACM Symposium on Theory of Computing (STOC 2009)*. pp. 631–638 (2009). <https://doi.org/10.1145/1536414.1536500>
7. Dean, A.: A layout algorithm for bar-visibility graphs on the Möbius band. In: Marks, J. (ed.) *Graph Drawing (GD 2000)*. Lecture Notes in Computer Science, vol. 1984, pp. 350–359. Springer (2000). https://doi.org/10.1007/3-540-44541-2_33
8. Dean, A., Evans, W., Gethner, E., Laison, J., Safari, M.A., Trotter, W.: Bar k -visibility graphs. *J. Graph Algorithms Appl.* **11**(1), 45–59 (2007). <https://doi.org/10.7155/jgaa.00136>
9. Diestel, R.: *Graph Theory*, 4th Edition, Graduate texts in mathematics, vol. 173. Springer (2012)

10. Duchet, P., Hamidoune, Y., Vergnas, M.L., Meyniel, H.: Representing a planar graph by vertical lines joining different levels. *Discret. Math.* **46**(3), 319–321 (1983). [https://doi.org/10.1016/0012-365X\(83\)90128-0](https://doi.org/10.1016/0012-365X(83)90128-0)
11. Even, S., Tarjan, R.E.: Computing an *st*-numbering. *Theoretical Computer Science* **2**, 436–441 (1976). [https://doi.org/10.1016/0304-3975\(76\)90086-4](https://doi.org/10.1016/0304-3975(76)90086-4)
12. de Fraysseix, H., Ossona de Mendez, P., Pach, J.: Representation of planar graphs by segments. *Intuitive Geometry* **63**, 109–117 (1991)
13. Fraysseix, H.d., Pach, J., Pollack, R.: How to draw a planar graph on a grid. *Combinatorica* **10**, 41–51 (1990). <https://doi.org/10.1007/BF02122694>
14. Gonçalves, D., Isenmann, L., Pennarun, C.: Planar graphs as L-intersection or L-contact graphs. In: *SIAM Symposium on Discrete Algorithms (SODA 2018)*. pp. 172–184 (2018). <https://doi.org/10.1137/1.9781611975031.12>
15. Gross, J.L., Tucker, T.W.: *Topological Graph Theory*. John Wiley and Sons (1987)
16. Hutchinson, J.: A note on rectilinear and polar visibility graphs. *Discret. Appl. Math.* **148**(3), 263–272 (2005). <https://doi.org/10.1016/j.dam.2004.12.004>
17. Lempel, A., Even, S., Cederbaum, I.: An algorithm for planarity testing of graphs. In: *Theory of Graphs, International Symposium Rome 1966*. pp. 215–232. Gordon and Breach (1967)
18. Lengauer, T.: *Combinatorial Algorithms for Integrated Circuit Layout*. Teubner/Wiley & Sons, Stuttgart/Chicester (1990)
19. Middendorf, M., Pfeiffer, F.: The max clique problem in classes of string-graphs. *Discrete Mathematics* **108**(1-3), 365–372 (1992). [https://doi.org/10.1016/0012-365X\(92\)90688-C](https://doi.org/10.1016/0012-365X(92)90688-C)
20. Mohar, B., Rosenstiehl, P.: Tessellation and visibility representations of maps on the torus. *Discret. Comput. Geom.* **19**(2), 249–263 (1998). <https://doi.org/10.1007/PL00009344>
21. Otten, R., van Wijk, J.: Graph representations in interactive layout design. In: *IEEE International Symposium on Circuits and Systems*. pp. 914–918. New York (1978)
22. Ripphausen-Lipa, H., Wagner, D., Weihe, K.: The vertex-disjoint Menger problem in planar graphs. *SIAM J. Comput.* **26**(2), 331–349 (1997). <https://doi.org/10.1137/S0097539793253565>
23. Rosenstiehl, P., Tarjan, R.E.: Rectilinear planar layouts and bipolar orientation of planar graphs. *Discrete Computational Geometry* **1**, 343–353 (1986). <https://doi.org/10.1007/BF02187706>
24. Tamassia, R., Tollis, I.: A unified approach to visibility representations of planar graphs. *Discrete Computational Geometry* **1**, 321–341 (1986). <https://doi.org/10.1007/BF02187705>
25. Tamassia, R., Tollis, I.: Planar grid embedding in linear time. *IEEE Transactions on Circuits and Systems* **36**(9), 1230–1234 (1989)
26. Tamassia, R., Tollis, I.: Representations of graphs on a cylinder. *SIAM J. Discret. Math.* **4**(1), 139–149 (1991). <https://doi.org/10.1137/0404014>
27. Wismath, S.: Characterizing bar line-of-sight graphs. In: *ACM Symposium on Computational Geometry (SoCG '85)*. pp. 147–152 (1985). <https://doi.org/10.1145/323233.323253>