Pixel and Voxel Representations of Graphs[⋆]

M. J. Alam[1], T. Bläsius[2], I. Rutter[2], T. Ueckerdt[2], and A. Wolff[3]

[1] University of Arizona
[2] Karlsruhe Institute of Technology
[3] Universität Würzburg

Abstract. We study contact representations for graphs, which we call *pixel representations* in 2D and *voxel representations* in 3D. Our representations are based on the unit square grid whose cells we call pixels in 2D and voxels in 3D. Two pixels are adjacent if they share an edge, two voxels if they share a face. We call a connected set of pixels or voxels a *blob*. Given a graph, we represent its vertices by disjoint blobs such that two blobs contain adjacent pixels or voxels if and only if the corresponding vertices are adjacent. We are interested in the size of a representation, which is the number of pixels or voxels it consists of.
We first show that finding minimum-size representations is NP-complete. Then, we bound representation sizes needed for certain graph classes. In 2D, we show that, for k-outerplanar graphs with n vertices, $\Theta(kn)$ pixels are always sufficient and sometimes necessary. In particular, outerplanar graphs can be represented with a linear number of pixels, whereas general planar graphs sometimes need a quadratic number. In 3D, $\Theta(n^2)$ voxels are always sufficient and sometimes necessary for any n-vertex graph. We improve this bound to $\Theta(n \cdot \tau)$ for graphs of treewidth τ and to $O((g+1)^2 n \log^2 n)$ for graphs of genus g. In particular, planar graphs admit representations with $O(n \log^2 n)$ voxels.

1 Introduction

In Tutte's landmark paper "How to draw a graph", he introduces barycentric coordinates as a tool to draw triconnected planar graphs. Given the positions of the vertices on the outer face (which must be in convex position), the positions of the remaining vertices are determined as the solutions of a set of equations. While the solutions can be approximated numerically, and symmetries tend to be reflected nicely in the resulting drawings, the ratio between the lengths of the longest edge and the shortest edge is exponential in many cases. This deficiency triggered research directed towards drawing graphs on grids of small size in both 2D and 3D for different graph drawing paradigms; Brandenburg et al. [11] listed this as an important open problem. In *straight-line grid drawings*, the vertices are at integer grid points and the edges are drawn as straight-line segments. Both, Schnyder [35] and de Fraysseix et al. [28], gave algorithms for drawing any n-vertex planar graph on a grid of size $O(n) \times O(n)$. There has also been research towards drawing subclasses of planar graphs on small-area grids. For example, any n-vertex outerplanar graph can be drawn in area $O(n^{1.48})$ [17]. Similar research has also been done for other graph drawing problems, such as *polyline drawings*, where edges

[⋆] This work was started at the 2014 Bertinoro Workshop on Graph Drawing. We thank the organizers for creating an inspiring atmosphere and Sue Whitesides for suggesting the problem. A.W. acknowledges support by the ESF EuroGIGA project GraDR (DFG grant Wo 758/5-1).

can have bends [8], *orthogonal drawings*, where edges are polylines consisting of only axis-aligned segments [8, 15], and for drawing graphs in 3D [19, 33, 34]

A *bar visibility representation* [36] draws a graph in a different way: the vertices are horizontal segments and the edges are realized by vertical line-of-sights between corresponding segments. Improving earlier results, Fan et al. [22] showed that any planar graph admits a visibility representation of size $(\lfloor 4n/3 \rfloor - 2) \times (n-1)$. Generalized visibility representations for non-planar graphs have been considered in 2D [12, 21], and in 3D [10]. In all these and many subsequent papers, the size of a drawing is measured as the area or volume of the bounding box.

Yet another approach to drawing graphs are the so-called *contact representations*, where vertices are interior-disjoint geometric objects such as lines, curves, circles, polygons, polyhedra, etc. and edges correspond to pairs of objects touching in some specified way. An early work by Koebe [31] represents planar graphs with touching disks in 2D. Any planar graph can also be represented by contacts of triangles [27], by side-to-side contacts of hexagons [20] and of axis-aligned T-shape polygons [2,27]. 2D-contact representations of graphs with curves [29], line-segments [26], L-shapes [16], homothetic triangles [3], squares and rectangles [14, 23] have also been studied. Of particular interest are the so-called *VCPG-representations* [1], where vertices are represented by interior-disjoint paths in the plane square grid and edges are contacts between an endpoint of one path and an interior point of another.

Contact representations in 3D allow us to visualize non-planar graphs, but little is known about contact representations in 3D: Any planar graph can be represented by contacts of cubes [24], and by face-to-face contact of boxes [13, 37]. Contact representations of complete graphs and complete bipartite graphs in 3D have been studied using spheres [5, 30], cylinders [4], and tetrahedra [38]. In 3D as well as in 2D, the complexity of a contact representation is usually measured in terms of the *polygonal complexity* (i.e., the number of corners) of the objects used in the representation.

In this paper, in contrast, we are interested in "building" graphs, and so we aim at minimizing the cost of the building material—think of unit-size Lego-like blocks that can be connected to each other face-to-face. We represent each vertex by a connected set of building blocks, which we call a *blob*. If two vertices are adjacent, the blob of one vertex contains a block that is connected (face-to-face) to a block in the blob of the other. The blobs of two non-adjacent vertices are not connected. We call the building blocks *pixels* in 2D and *voxels* in 3D. Accordingly, the 2D and 3D variants of such representations are called *pixel* and *voxel* representations, respectively. We define the *size* of a pixel or voxel representation to be the total number of boxes it consists of. (We use the generic term *box* to denote either pixel or voxel when the dimension is not important.)

Although pixel representations can be thought of as a generalization of VCPG-representations, where grid subgraphs, instead of grid paths are used, to the best of our knowledge, minimizing or bounding the size of such representations has not been studied so far, neither in 2D nor in 3D.

Our Contribution. We first investigate the complexity of our problem: finding minimum-size representations turns out to be NP-complete (Section 2). Then, we give lower and upper bounds for the sizes of 2D- and 3D-representations for certain graph classes:

- In 2D, we show that, for k-outerplanar graphs with n vertices, $\Theta(kn)$ pixels are always sufficient and sometimes necessary (see Section 3). In particular, outerplanar graphs can be represented with a linear number of pixels, whereas general planar graphs sometimes need a quadratic number.
- In 3D, $\Theta(n^2)$ voxels are always sufficient and sometimes necessary for any n-vertex graph (see Section 4). We improve this bound to $\Theta(n \cdot \tau)$ for graphs of treewidth τ and to $O((g+1)^2 n \log^2 n)$ for graphs of genus g. In particular, n-vertex planar graphs admit voxel representations with $O(n \log^2 n)$ voxels.

2 Complexity

First, we show that it is NP-hard to compute minimum-size pixel representations. We reduce from the problem of deciding if a planar graph of maximum degree 4 has a grid drawing with edges of length 1. This problem is known to be NP-hard [6]. The hardness proof still works if the angles between adjacent edges are specified. Note that specifying the angles also prescribes the circular order of edges around vertices (up to reversal). We give a sketch of the hardness proof here, details are in Appendix A.

Theorem 1. *It is NP-complete to minimize the size of a pixel representation of a planar graph.*

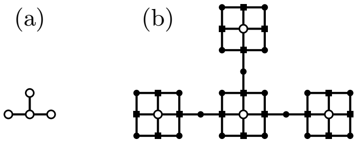

Fig. 1: (a) The graph G with prescribed angles between edges, and edges drawn with length 1. (b) The graph H drawn with edge length 1.

Proof sketch. Let G be a planar graph of maximum degree 4 with prescribed angles between edges. Construct a graph H by replacing each vertex by a wheel of five vertices so that the angles between the edges are respected, and subdividing each edge except the ones incident to the wheel-centers. Then G has a grid drawing with edge length 1 if and only if H has a representation where each vertex is a pixel. Indeed, from a grid drawing of G one can obtain a drawing of H where two vertices have distance 1 if and only if they are adjacent; see Fig. 1. Represent each vertex v of H by a pixel with v at its center. Conversely, if H has a representation where each vertex is a pixel, then for each vertex v of G, the subdivided wheel is a 3×3 square. Placing each vertex v at the center of the square and scaling by $1/4$ yields the grid drawing of G. □

Next, we reduce computing minimum-size pixel representations to computing minimum-size voxel representations. In our reduction, we build a rigid structure around the given graph that forces the given graph to be drawn in a single plane; see Appendix A.

Theorem 2. *It is NP-complete to minimize the size of a voxel representation of a graph.*

3 Lower and Upper Bounds in 2D

Here we only consider planar graphs since only planar graphs admit pixel representations. Let G be a planar graph with fixed plane embedding $\mathcal{E}$. The embedding $\mathcal{E}$ is 1-*outerplane* (or simply outerplane) if all vertices are on the outer face. It is *k-outerplane* if removing all vertices on the outer face yields a $(k-1)$-outerplane embedding. A graph G is *k-outerplanar* if it admits a k-outerplane embedding but no k'-outerplane embedding for $k' < k$. Note that $k \in O(n)$, where n is the number of vertices of G.

In Section 3.1, we show that pixel representations of an n-vertex k-outerplanar graph sometimes requires $\Omega(kn)$ pixels. As the number of pixels is a lower bound for the area consumption, this strengthens a result by Dolev et al. [18] that says that orthogonal drawings of planar graphs of maximum degree 4 and *width* w sometimes require $\Omega(wn)$ area. As we will see later, width and k-outerplanarity are very similar concepts.

In Section 3.2, we show that $O(kn)$ area and thus using $O(kn)$ pixels is also sufficient. We use a result by Dolev et al. [18] who proved that any n-vertex planar graph of maximum degree 4 and width w admits a planar orthogonal drawing of area $O(wn)$. The main difficulty is to extend their result to general planar graphs.

3.1 Lower Bound

Let G be a k-outerplanar graph with a pixel representation Γ. Note that a pixel representation Γ induces an embedding of G. Let Γ induce a k-outerplane embedding of G, which we call a *k-outerplane pixel representation* for short. We claim that the width and the height of Γ are at least $2k-1$. For $k=1$ this is trivial as every (non-empty) graph requires width and height at least 1. For $k \geq 2$, let $V_{\text{ext}} = \{v_1, \ldots, v_\ell\}$ be the set of vertices incident to the outer face of Γ. Removing V_{ext} from G yields a $(k-1)$-outerplane graph G' with corresponding pixel representation Γ'. By induction, Γ' requires width and height $2(k-1)-1$. As the representation of V_{ext} in Γ encloses the whole representation Γ' in its interior, the width and the height of Γ are at least two units larger than the width and the height of Γ', respectively.

Clearly, the number of pixels required by the vertices in V_{ext} is at least the perimeter of Γ (twice the width plus twice the height minus 4 for the corners, which are shared) and thus at least $8k-8$. After removing the vertices in V_{ext}, the new vertices on the outer face require $8(k-1)-8$ pixels, and so on. Thus, Γ requires overall at least $\sum_{i=1}^{k}(8i-8) = 4k^2-4k$ pixels, which gives the following lemma.

Lemma 1. *Any k-outerplane pixel representation has size at least* $4k^2-4k$*.*

There are k-outerplanar graphs with n vertices such that $k \in \Theta(n)$. For example, the nested triangle graph with $2k$ triangles (see Fig. 2) has $n=6k$ vertices and is k-outerplanar for $k \geq 2$. Let G be a graph with c connected components each of which is k-outerplanar and has $\Theta(k)$ vertices. Then each connected component requires $4k^2-4k$ pixels (due to Lemma 1) and thus we need at least $(4k^2-4k)c$ pixels in total. As G has $n = \Theta(kc)$ vertices, we get $(4k^2-4k)c \in \Theta(kn)$, which proves the following.

Theorem 3. *Some k-outerplanar graphs require* $\Omega(kn)$*-size pixel representations.*

3.2 Upper Bound

In the following two lemmas, we first show how to construct a pixel representation from a given orthogonal drawing and that taking minors does not heavily increase the number of pixels we need. Both lemmas aim at extending a result of Dolev et al. [18] on orthogonal drawings of planar graphs with maximum degree 4 to pixel representations of general planar graphs. As we re-use both lemmas in the 3D case (Section 4), we state them in the general d-dimensional setting.

Lemma 2. *Let G be a graph with n vertices, m edges, and an orthogonal drawing of total edge length ℓ in d-dimensional space. Then G admits a d-dimensional representation of size* $2\ell+n-m$*.*

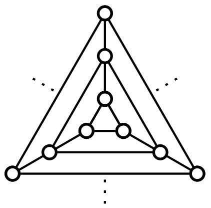

Fig. 2: A nested triangle graph of outerplanarity $\Omega(n)$.

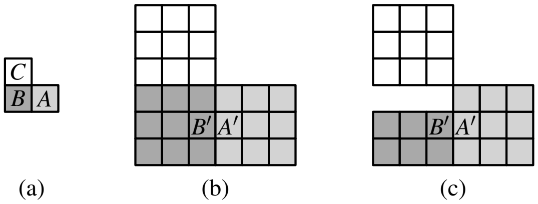

Fig. 3: Constructing a representation of a minor with asymptotically the same number of blocks.

Proof. We first scale the given drawing Γ of G by a factor of 2 and subdivide the edges of G such that every edge has length 1. Denote the resulting graph by G' and its drawing by Γ'. An edge e of length ℓ_e in Γ is represented by a path with $2\ell_e - 1$ internal vertices (the subdivision vertices). Thus, the total number of subdivision vertices is $2\ell - m$. Due to the scaling, non-adjacent vertices in G' have distance greater than 1 in Γ' (adjacent vertices have distance 1). Thus, representing every vertex v by the box having v as center yields a representation of G' with $2\ell + n - m$ boxes (one box per vertex of G'). If we assign the boxes representing subdivision vertices to one of the endpoints of the corresponding edge, we get a representation of G with $2\ell + n - m$ boxes. □

Lemma 3. *Let G be a graph that has a d-dimensional representation of size b. Every minor of G admits a d-dimensional representation of size at most $3^d b$.*

Proof. Let H be a minor obtained from G by first deleting some edges, then deleting isolated vertices, and finally contracting edges. We start with the representation Γ of G using b boxes and scale it by a factor of 3. This yields a representation 3Γ using $3^d b$ boxes. Then we modify 3Γ, without adding boxes, to represent the minor H. For convenience, we consider the 2D case; the case $d > 2$ works analogously.

Let uv be an edge in G that is deleted. In 3Γ we delete every pixel in the representation of u that touches a pixel of the representation of v. We claim that this neither destroys the contact of u with any other vertex nor does it disconnect the shape representing u. Consider a single pixel B in Γ. In 3Γ it is represented by a square of 3×3 pixels *belonging* to B. If B is in contact to another pixel A in Γ, then there is a pair of pixels A' and B' in 3Γ such that A' and B' are in contact, while all other pixels that touch A' and B' belong to A and B, respectively; see Figs. 3a and 3b. Assume that we remove in 3Γ all pixels belonging to B that are in contact to pixels belonging to another pixel C touching B in Γ; see Fig. 3c. Obviously, this does not effect the contact between A' and B'. Moreover, the remaining pixels belonging to B form a connected blob. The above claim follows immediately.

Removing isolated vertices can be done by simply removing their representation. Moreover, contracting an edge uv into a vertex w can be done by merging the blobs representing u and v into a single blob representing w. This blob is obviously connected and touches the blob of another vertex if and only if either u or v touch this vertex. □

Now let G be a k-outerplanar graph. Applying the algorithm of Dolev et al. [18] yields an orthogonal drawing of total length $O(wn)$, where w is the *width* of G. The width w of G is the maximum number of vertices contained in a shortest path from an arbitrary vertex of G to a vertex on the outer face. Given the orthogonal drawing, Lemma 2

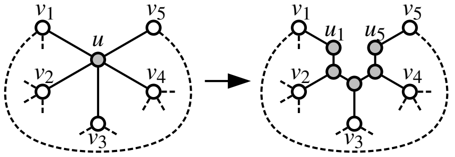

Fig. 4: Replacement of high-degree vertices while preserving k-outerplanarity.

gives us a pixel representation of G. There are, however, two issues. First, k and w are not the same (e.g., subdividing edges increases w but not k). Second, G does not have maximum degree 4, thus we cannot simply apply the algorithm of Dolev et al. [18].

Concerning the first issue, we note that the algorithm of Dolev et al. exploits that G has width w only to find a special type of separator [18, Theorem 1]. For this, it is sufficient that G is a subgraph of a graph of width w (not necessarily with maximum degree 4; in fact Dolev et al. triangulate the graph before finding the separator).

Lemma 4. *Every k-outerplanar graph has a planar supergraph of width $w = k$.*

Proof. Let G be a graph with a k-outerplane embedding. Iteratively deleting the vertices on the outer face gives us a sequence of deletion phases. For each vertex v, let k_v be the phase in which v is deleted. Note that the maximum over all values of k_v is exactly k. For any vertex v, either $k_v = 1$ or there is a vertex u with $k_u = k_v - 1$ such that u and v are incident to a common face. Thus, there is a sequence $v_1, \dots, v_{k_v}$ of k_v vertices such that (i) $v_1 = v$, (ii) v_{k_v} lies on the outer face, and (iii) v_i, v_{i+1} are incident to a common face. If the graph G was triangulated, this would yield a path containing k_v vertices from v to a vertex on the outer face. Thus, triangulated k-outerplanar graphs have width $w = k$.

It remains to show that G can be triangulated without increasing k_v for any vertex v. Consider a face f and let u be the vertex incident to f for which k_u is minimal. Let $v \neq u$ be any other vertex incident to f. Adding the edge uv clearly does not increase the value k_x for any vertex x. We add edges in this way until the graph is triangulated. Alternatively, we can use a result of Biedl [7] to triangulate G. Note that we do not need to triangulate the outer face of G. Hence, we do not increase the outerplanarity. □

To solve the second issue (the k-outerplanar graph G not having maximum degree 4), we construct a graph G' such that G is a minor of G', G' is k-outerplanar, and G' has maximum degree 4. Then, (due to Lemma 4) we can apply the algorithm of Dolev et al. [18] to G'. Next, we apply Lemma 2 to the resulting drawing to get a representation of G' with $O(kn)$ pixels. As G is a minor of G', Lemma 3 yields a representation of G that, too, requires $O(kn)$ pixels.

Theorem 4. *Every k-outerplanar n-vertex graph has a size $O(kn)$ pixel representation.*

Proof. Let G be a k-outerplanar graph. After the above considerations, it remains to construct a k-outerplanar graph G' with maximum degree 4 such that G is a minor of G'. Let u be a vertex with $\deg(u) > 4$. We replace u with a path of length $\deg(u)$ and connect each neighbor of u to a unique vertex of this path. This can be done maintaining a plane embedding. We now show that the resulting graph remains k-outerplanar.

Consider a vertex u on the outer face with neighbors $v_1, \dots, v_\ell$. Assume the neighbors appear in that order around u such that v_1 is the counter-clockwise successor of u

on the outer face; see Fig. 4. We replace u with the path $u_1,\dots,u_\ell$ and connect u_i to v_i for $1 \le i \le \ell$. Call the resulting graph G_u. Note that all u_i in G_u are incident to the outer face. Thus, if G was k-outerplanar, G_u is also k-outerplanar. Moreover, the degrees of the new vertices do not exceed 4 (actually not even 3), and G is a minor of G_u—one can simply contract the inserted path to obtain G.

We can basically apply the same replacement if u is not incident to the outer face. Assume that we delete u in phase k_u if we iteratively delete vertices incident to the outer face. When replacing u with the vertices $u_1,\dots,u_\ell$, we have to make sure that all these vertices get deleted in phase k_u. Let f be a face incident to u that is merged with the outer face after $k_u - 1$ deletion phases (such a face must exist, otherwise u is not deleted in phase k_u). We apply the same replacement as for the case where u was incident to the outer face, but this time we ensure that the new vertices u_i are incident to the face f. Thus, after $k_u - 1$ deletion phases they are all incident to the outer face and thus they are deleted in phase k_u. Hence, the resulting graph G_u is k-outerplanar. Again the new vertices have degree at most 3 and G is obviously a minor of G_u. Iteratively applying this kind of replacement for ever vertex u with $\deg(u) > 4$ yields the claimed graph G'.

The corresponding drawing can then be obtained as follows. Since G' has a supergraph of width $w = k$ by Lemma 4, and G' has maximum degree 4, we use the algorithm of Dolev et al. [18] to obtain a drawing of G' with area (and hence total edge length) $O(nk)$. By Lemma 2, we thus obtain a representation of G' with $O(nk)$ pixels. Since G is a minor of G', Lemma 3 yields a representation of G with $O(nk)$ pixels. □

4 Representations in 3D

In this section, we consider voxel representations. We start with some basic considerations showing that every n-vertex graph admits a representation with $O(n^2)$ voxels. Note that $\Omega(n^2)$ is obviously necessary for K_n as every edge corresponds to a face-to-face contact and every voxel has at most 6 such contacts. We improve on this simple general result in two ways. First, we show that n-vertex graphs with treewidth at most τ admit voxel representations of size $O(n \cdot \tau)$ (see Section 4.1). Second, for n-vertex graphs with genus at most g, we obtain representations with $O(g^2 n \log^2 n)$ voxels (see Section 4.2).

Theorem 5. *Any n-vertex graph admits a voxel representation of size $O(n^2)$.*

Proof. Let G be a graph with vertices $v_1,\dots,v_n$. Vertex v_i $(i = 1,\dots,n)$ is represented by three cuboids (see Fig. 5a), namely a vertical cuboid consisting of the voxels centered at the points $(2i,2,0),(2i,3,0),\dots,(2i,2n,0)$, a horizontal cuboid consisting of the voxels centered at $(2,2i,2),(3,2i,2),\dots,(2n,2i,2)$, and the voxel centered at $(2i,2i,1)$. This yields a representation where every vertex is a connected blob and no two blobs are in contact. Moreover, for every pair of vertices v_i and v_j, there is a voxel of v_i at $(2i,2j,0)$ and a voxel of v_j at $(2i,2j,2)$ and no voxel between them at $(2i,2j,1)$. Thus, one can easily represent an arbitrary edge (v_i,v_j) by extending the representation of v_i to also contain $(2i,2j,1)$; see Fig. 5b. Clearly, this representation consists of $O(n^2)$ voxels. □

4.1 Graphs of Bounded Treewidth

Let $G = (V,E)$ be a graph. A *tree decomposition* of G is a tree T where each node μ in T is associated with a *bag* $X_\mu \subseteq V$ such that: (i) for each $v \in V$, the nodes of T whose

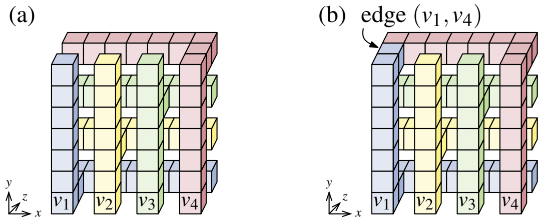

Fig. 5: (a) The basic contact representation without any contacts between vertices. (b) If v_1 and v_4 are adjacent, it suffices to add a single voxel to the representation of v_1 (or to that of v_4).

bags contain v form a connected subtree, and (ii) for each edge $uv \in E$, T contains a node μ such that $u, v \in X_\mu$.

Note that we use (lower case) Greek letters for the nodes of T to distinguish them from the vertices of G. The *width* of the tree decomposition is the maximum bag size minus 1. The *treewidth* of G is the minimum width over all tree decompositions of G. A tree decomposition is *nice* if T is a rooted binary tree, where for every node μ:

- μ is a leaf and $|X_\mu| = 1$ (*leaf node)*, or
- μ has a single child η with $X_\mu \subseteq X_\eta$ and $|X_\mu| = |X_\eta| - 1$ (*forget node*), or
- μ has a single child η with $X_\eta \subseteq X_\mu$ and $|X_\mu| = |X_\eta| + 1$ (*introduce node*), or
- μ has two children η and κ with $X_\mu = X_\eta = X_\kappa$ (*join node*).

Any tree decomposition can be transformed (without increasing its width) into a nice tree decomposition such that the resulting tree T has $O(n)$ nodes, where n is the number of vertices of G [9]. This transformation can be done in linear time. Thus, we can assume any tree decomposition to be a nice tree decomposition with a tree of size $O(n)$.

Lemma 5. *Let T be a nice tree decomposition of a graph G. The edges of G can be mapped to the nodes of T such that every edge uv of G is mapped to a node μ with $u, v \in X_\mu$ and the edges mapped to each node μ form a star.*

Proof. We say that a node μ *represents* the edge uv if uv is mapped to μ. Consider a node μ during a bottom-up traversal of T. We want to maintain the invariant that, after processing μ, all edges between vertices in X_μ are represented by μ or by a descendant of μ. This ensures that every edge is represented by at least one node. Every edge can then be mapped to one of the nodes representing it.

If μ is a leaf, it cannot represent an edge as $|X_\mu| = 1$. If μ is a forget node, it has a child η with $X_\mu \subseteq X_\eta$. Thus, by induction, all edges between vertices in X_μ are already represented by descendants of μ. If μ is an introduce node, it has a child η and $X_\mu = X_\eta \cup \{u\}$ for a vertex u of G. By induction, all edges between nodes in X_η are already represented by descendants of μ. Thus, μ only needs to represent the edges between the new node u and other nodes in X_μ. Note that these edges form a star with center u. Finally, if μ is a join node, no edge needs to be represented by μ (by the same argument as for forget nodes). This concludes the proof. □

We obtain a small voxel representation roughly as follows. We start with a "2D" voxel representation of the tree T, that is, all voxel centers lie in the x–y plane. We take $\tau + 1$ copies of this representation and place them in different layers in 3D space. We then assign to each vertex v of G a piece of this layered representation such that its piece contains all nodes of T that include v in their bags. For an edge uv, let μ be the node to which uv is mapped by Lemma 5. By construction, the representation of μ occurs multiple times representing u and v in different layers. To represent uv, we only have to connect the representations of u and v. As it suffices to represent a star for each node μ in this way, the number of voxels additionally used for these connections is small.

Theorem 6. *Any n-vertex graph of treewidth τ has a voxel representation of size $O(n\tau)$.*

Proof. Let G be an n-vertex graph of treewidth τ. During our construction, we will get some contacts between the blobs of vertices that are actually not adjacent in G. As G is a minor of the graph that we represent this way, we can use Lemma 3 to get a representation of G. Let T be a nice tree decomposition of G. As a tree, T is outerplanar and, hence, admits a pixel representation Γ with $O(n)$ pixels (by Theorem 4). Let $\Gamma_1,\dots,\Gamma_k$ be voxel representations corresponding to Γ with z-coordinates $1,\dots,k=\tau+1$.

For a vertex v of G, we denote by $\Gamma_i(v)$ the sub-representation of Γ_i induced by the nodes of T whose bags contain v. Now let $c\colon V\to\{1,\dots,k\}$ be a k-coloring of G with color set $\{1,\dots,k\}$ such that no two vertices sharing a bag have the same color. Such a coloring can be computed by traversing T bottom up, assigning in every introduce node μ a color to the new vertex that is not already used by any other vertex in X_μ. As a basis for our construction, we represent each vertex v of G by the sub-representation $\Gamma_{c(v)}(v)$.

So far, we did not represent any edge of G. Our construction, however, has the following properties: (i) it uses $O(nk)$ voxels. (ii) every vertex is a connected set of voxels. (iii) for every node μ of T, there is a position (x_μ,y_μ) in the plane such that, for every vertex $v\in X_\mu$, the voxel at $(x_\mu,y_\mu,c(v))$ belongs to the representation of v. Scaling the representation by a factor of 2 ensures that this is not the only voxel for v and that v is not disconnected if this voxel is removed (or reassigned to another vertex).

By Lemma 5 it suffices to represent for every node μ edges between vertices in X_μ that form a star. Let u be the center of this star. We simply assign the voxels centered at $(x_\mu,y_\mu,1),\dots,(x_\mu,y_\mu,k)$ to the blob of u. This creates a contact between u and every other vertex $v\in X_\mu$ (by the above property that the voxel $(x_\mu,y_\mu,c(v))$ belonged to v before). Finally, we apply Lemma 3 to get rid of unwanted contacts. The resulting representation uses $O(nk)$ voxels, which concludes the proof. □

Note that cliques of size k require $\Omega(k^2)$ voxels. Taking the disjoint union of n/k such cliques yields graphs with n vertices requiring $\Omega(nk)$ voxels. Note that these graphs have treewidth $\tau=k-1$. Thus, the bound of Theorem 6 is asymptotically tight.

Theorem 7. *Some n-vertex graphs of treewidth τ require $\Omega(n\tau)$ voxels.*

4.2 Graphs of Bounded Genus

Since planar graphs (genus 0) have treewidth $O(\sqrt{n})$ [25], we can obtain a voxel representation of size $O(n^{1.5})$ for any planar graph, from Theorem 6. Next, we improve this bound to $O(n\log^2 n)$ by proving a more general result for graphs of bounded genus. Recall that we used known results on orthogonal drawings with small area to obtain small pixel representations in Section 3.2. Here we follow a similar approach (re-using Lemmas 2 and 3), now allowing the orthogonal drawing we start with to be non-planar.

We obtain small voxel representations by first showing that it is sufficient to consider graphs of maximum degree 4: we replace higher-degree vertices by connected

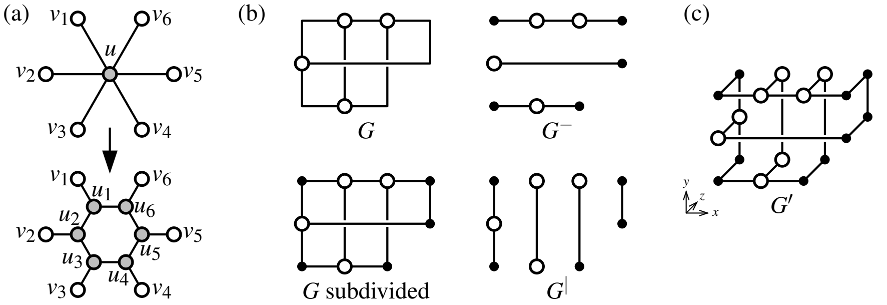

Fig. 6: Constructing voxel representations for bounded-genus graphs: (a) replacing high-degree vertices while preserving the genus, (b) subdividing and decomposing a graph according to a non-planar orthogonal drawing with small area, and (c) constructing a 3D drawing with small total edge length from the decomposition in (b).

subgraphs as in the proof of Theorem 4. Then we use a result of Leiserson [32] who showed that any graph of genus g and maximum degree 4 admits a 2D orthogonal drawing of area $O((g+1)^2 n \log^2 n)$, possibly with edge crossings. The area of an orthogonal drawing is clearly an upper bound for its total edge length. Finally we turn the pixels into voxels and use the third dimension to get rid of the crossings without using too many additional voxels.

Theorem 8. *Every n-vertex graph of genus g admits a voxel representation of size* $O((g+1)^2 n \log^2 n)$.

Proof. Let G be an n-vertex graph, and let u be a vertex of degree $\ell > 4$. Assume G to be embedded on a surface of genus g, and let $v_1, \dots, v_\ell$ be the neighbors of u appearing in that order around u (with respect to the embedding). We replace u with the cycle $u_1, \dots, u_\ell$ and connect u_i to v_i for $1 \le i \le \ell$; see Fig. 6a. Clearly, the new vertices have degree 3 and the genus of the graph has not increased. Applying this modification to every vertex of degree at least 5 yields a graph G_4 of maximum degree 4 and genus g. Moreover, G is a minor of G_4 as one can undo the cycle replacements by contracting all edges in the cycles. Thus, we can transform a voxel representation of G_4 into a voxel representation of G by applying Lemma 3.

We claim that the number n_4 of vertices in G_4 is linear in n. Indeed, if m denotes the number of edges in G, then we have $n_4 \le n + 2m$. Moreover, we can assume without loss of generality that $g \in O(n)$ (otherwise Theorem 5 already gives a better bound). This implies that $m \in O(n)$ and hence, $n_4 \in O(n)$, as we claimed.

We thus assume that G has maximum degree 4. Then G has a (possibly non-planar) orthogonal drawing Γ of total edge length $O(g^2 n \log^2 n)$ [32]. We modify G and Γ as follows. For every bend on an edge e in Γ, we subdivide the edge e once yielding a partition of the edges of the subdivided graph into horizontal and vertical edges. We obtain a graph G' from this subdivision of G by replacing every vertex v by two adjacent vertices v_1 and v_2, and connecting v_1 and w_1 (respectively v_2 and w_2) by an edge if v and w are connected by a horizontal (respectively vertical edge); see Fig. 6b.

We draw G' in 3D space by using the drawing Γ and setting for every vertex v the z-coordinate of v_1 and v_2 to 0 and 1, respectively. The x- and y-coordinates of vertices and edges are the same as in Γ; see Fig. 6b. Note that G is a minor of G': we obtain G

from G' by contracting (i) the edge v_0v_1 for every vertex v and (ii) any subdivision vertex. Asymptotically, the total edge length of Γ' is the same as that of Γ, that is, $O((g+1)^2 n \log^2 n)$. By Lemma 2, we turn Γ' into a voxel representation of G' and, by Lemma 3, into a voxel representation of G with size $O((g+1)^2 n \log^2 n)$. □

5 Conclusion

In this paper, we have studied pixel representations and voxel representations of graphs, where vertices are represented by disjoint blobs (that is, connected sets of grid cells) and edges correspond to pairs of blobs with face-to-face contact. We have shown that it is NP-complete to minimize the number of pixels or voxels in such representations. Does this problem admit an approximation algorithm?

We have shown that $O((g+1)^2 n \log^2 n)$ voxels suffice for any n-vertex graph of genus g. It remains open to improve this upper bound or to give a non-trivial lower bound. We believe that any planar graph admits a voxel representation of linear size.

References

1. Aerts, N., Felsner, S.: Vertex contact graphs of paths on a grid. In: Graph-Theoretic Concepts Comput. Sci. (WG'14). LNCS, vol. 8747, pp. 56–68 (2014)
2. Alam, M.J., Biedl, T., Felsner, S., Kaufmann, M., Kobourov, S., Ueckerdt, T.: Computing cartograms with optimal complexity. Discrete Comput. Geom. 50(3), 784–810 (2013)
3. Badent, M., Binucci, C., Giacomo, E.D., Didimo, W., Felsner, S., Giordano, F., Kratochvíl, J., Palladino, P., Patrignani, M., Trotta, F.: Homothetic triangle contact representations of planar graphs. In: Canad. Conf. Comput. Geom. (CCCG'07). pp. 233–236 (2007)
4. Bezdek, A.: On the number of mutually touching cylinders. Comb. Comput. Geom. 52, 121–127 (2005)
5. Bezdek, K., Reid, S.: Contact graphs of unit sphere packings revisited. J. Geom. 104(1), 57–83 (2013)
6. Bhatt, S.N., Cosmadakis, S.S.: The complexity of minimizing wire lengths in VLSI layouts. Inform. Process. Lett. 25(4), 263–267 (1987)
7. Biedl, T.: On triangulating k-outerplanar graphs. CoRR abs/1310.1845 (2013)
8. Biedl, T.C.: Small drawings of outerplanar graphs, series-parallel graphs, and other planar graphs. Discrete Comput. Geom. 45(1), 141–160 (2011)
9. Bodlaender, H.L.: Treewidth: Algorithmic techniques and results. In: Math. Foundat. Comput. Sci. (MFCS'97). LNCS, vol. 1295, pp. 19–36 (1997)
10. Bose, P., Everett, H., Fekete, S.P., Houle, M.E., Lubiw, A., Meijer, H., Romanik, K., Rote, G., Shermer, T.C., Whitesides, S., Zelle, C.: A visibility representation for graphs in three dimensions. J. Graph Algorithms Appl. 2(2) (1998)
11. Brandenburg, F., Eppstein, D., Goodrich, M., Kobourov, S., Liotta, G., Mutzel, P.: Selected open problems in graph drawing. In: Graph Drawing (GD'03). LNCS, vol. 2912, pp. 515–539 (2003)
12. Brandenburg, F.J.: 1-visibility representations of 1-planar graphs. J. Graph Algorithms Appl. 18(3), 421–438 (2014)
13. Bremner, D., Evans, W.S., Frati, F., Heyer, L.J., Kobourov, S.G., Lenhart, W.J., Liotta, G., Rappaport, D., Whitesides, S.: On representing graphs by touching cuboids. In: Graph Drawing (GD'12). LNCS, vol. 7704, pp. 187–198 (2012)
14. Buchsbaum, A.L., Gansner, E.R., Procopiuc, C.M., Venkatasubramanian, S.: Rectangular layouts and contact graphs. ACM Transactions on Algorithms 4(1) (2008)

15. Chan, T.M., Goodrich, M.T., Kosaraju, S.R., Tamassia, R.: Optimizing area and aspect ratio in straight-line orthogonal tree drawings. Comput. Geom. Theory Appl. 23(2), 153–162 (2002)
16. Chaplick, S., Kobourov, S.G., Ueckerdt, T.: Equilateral L-contact graphs. In: Graph-Theoretic Concepts Comput. Sci. (WG'13). LNCS, vol. 8165, pp. 139–151 (2013)
17. Di Battista, G., Frati, F.: Small area drawings of outerplanar graphs. Algorithmica 54(1), 25–53 (2009)
18. Dolev, D., Leighton, T., Trickey, H.: Planar embedding of planar graphs. Adv. Comput. Research 2, 147–161 (1984)
19. Dujmović, V., Morin, P., Wood, D.: Layered separators for queue layouts, 3d graph drawing and nonrepetitive coloring. In: Foundat. Comput. Sci. (FOCS'13). pp. 280–289 (2013)
20. Duncan, C.A., Gansner, E.R., Hu, Y.F., Kaufmann, M., Kobourov, S.G.: Optimal polygonal representation of planar graphs. Algorithmica 63(3), 672–691 (2012)
21. Evans, W., Kaufmann, M., Lenhart, W., Liotta, G., Mchedlidze, T., Wismath, S.: Bar 1-visibility graphs and their relation to other nearly planar graphs. CoRR abs/1312.5520 (2013)
22. Fan, J., Lin, C., Lu, H., Yen, H.: Width-optimal visibility representations of plane graphs. In: Int. Symp. Algorithms Comput. (ISAAC'07). LNCS, vol. 4835, pp. 160–171 (2007)
23. Felsner, S.: Rectangle and square representations of planar graphs. Thirty Essays on Geometric Graph Theory pp. 213–248 (2013)
24. Felsner, S., Francis, M.C.: Contact representations of planar graphs with cubes. In: Symp. Comput. Geom. (SoCG'11). pp. 315–320 (2011)
25. Fomin, F.V., Thilikos, D.M.: New upper bounds on the decomposability of planar graphs. J. Graph Theory 51(1), 53–81 (2006)
26. de Fraysseix, H., de Mendez, P.O.: Representations by contact and intersection of segments. Algorithmica 47(4), 453–463 (2007)
27. de Fraysseix, H., de Mendez, P.O., Rosenstiehl, P.: On triangle contact graphs. Comb. Prob. Comput. 3, 233–246 (1994)
28. de Fraysseix, H., Pach, J., Pollack, R.: How to draw a planar graph on a grid. Combinatorica 10(1), 41–51 (1990)
29. Hliněný, P.: Classes and recognition of curve contact graphs. J. Comb. Theory B 74(1), 87–103 (1998)
30. Hliněný, P., Kratochvíl, J.: Representing graphs by disks and balls (a survey of recognition-complexity results). Discrete Math. 229(1–3), 101–124 (2001)
31. Koebe, P.: Kontaktprobleme der konformen Abbildung. Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig. Math.-Phy. Kla. 88, 141–164 (1936)
32. Leiserson, C.E.: Area-efficient graph layouts (for VLSI). In: Foundat. Comput. Sci. (FOCS'80). pp. 270–281 (1980)
33. Pach, J., Thiele, T., Tóth, G.: Three-dimensional grid drawings of graphs. In: Graph Drawing (GD'97). LNCS, vol. 1353, pp. 47–51 (1997)
34. Patrignani, M.: Complexity results for three-dimensional orthogonal graph drawing. In: Graph Drawing (GD'05). LNCS, vol. 3843, pp. 368–379 (2005)
35. Schnyder, W.: Embedding planar graphs on the grid. In: Symp. Discrete Algorithms (SODA'90). pp. 138–148 (1990)
36. Tamassia, R., Tollis, I.G.: A unified approach a visibility representation of planar graphs. Discrete Comput. Geom. 1, 321–341 (1986)
37. Thomassen, C.: Interval representations of planar graphs. J. Comb. Theory B 40(1), 9–20 (1986)
38. Zong, C.: The kissing numbers of tetrahedra. Discrete Comput. Geom. 15(3), 239–252 (1996)

A NP-Completeness of Minimum-Size Pixel and Voxel Representations

Theorem 1. *It is NP-complete to minimize the size of a pixel representation of a planar graph.*

Proof. Clearly the corresponding decision problem is in NP, thus it remains to show NP-hardness. Let G be a planar graph of maximum degree 4 and assume that the angles between adjacent edges are prescribed. We define a graph H as follows (see Figs. 7a and 7b). First, replace every vertex by a wheel with five vertices such that the angles between the edges are respected. Second, subdivide every edge except those that are incident to the center of a wheel. We claim that G admits a grid drawing with edges of length 1 (respecting the prescribed angles) if and only if H admits a representation where every vertex is represented by exactly one pixel.

Assume G admits a grid drawing with edges of length 1. Scaling the drawing by a factor of 4 and suitably adding the new vertices and edges clearly yields a drawing of H with edges of length 1, such that two vertices have distance 1 only if they are adjacent; see Fig. 7b. For every vertex v of H, create a pixel P_v with v at its center (Fig. 7c). Clearly, for two adjacent vertices u and v in H, the pixels P_u and P_v touch as the edge uv has length 1 in the drawing of H. Moreover, two pixels P_u, P_v touch only if u, v have distance 1, hence only if u, v are adjacent. Thus this gives a pixel representation of H.

Conversely, assume H admits a representation such that every vertex v is represented by a single pixel. Obviously, the subdivided wheel of size 4 has a unique representation (up to symmetries) consisting of a square of 3×3 pixels. Consider two adjacent vertices u and v of G. Then there is a 3×3 square for u and one for v. As u and v adjacent in G, there must be a pixel representing the subdivision vertex on the edge uv in H that touches both 3×3 squares (of u and v) as in Fig. 7d. Thus, the straight line from the center of the square representing u to the center of the square representing v is horizontal or vertical and has length 4. Hence, we obtain a drawing of G with edges of length 4. Scaling by a factor of $1/4$ yields a grid drawing of G with edges of length 1. □

Theorem 2. *It is NP-complete to minimize the size of a voxel representation of a graph.*

Proof. Again, the corresponding decision problem is clearly in NP. To show NP-hardness, we reduce from the 2D case. To this end, we build a rigid structure called *cage* that forces the graph in which we are actually interested to be drawn in a single plane.

To simplify notation, we first prove for the 2-dimensional equivalent of a 3-dimensional cage that it actually is a rigid structure. We then extend this to 3D. The cage is basically the grid graph with a hole; see Fig. 8a. More precisely, the cage is defined

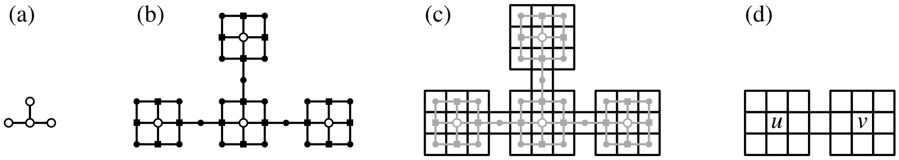

Fig. 7: (a) The graph G with prescribed angles between edges, edges drawn with length 1. (b) The graph H drawn with edge length 1. (c) Representation of H with unit squares. (d) representation of a subgraph of H corresponding to an edge uv in G.

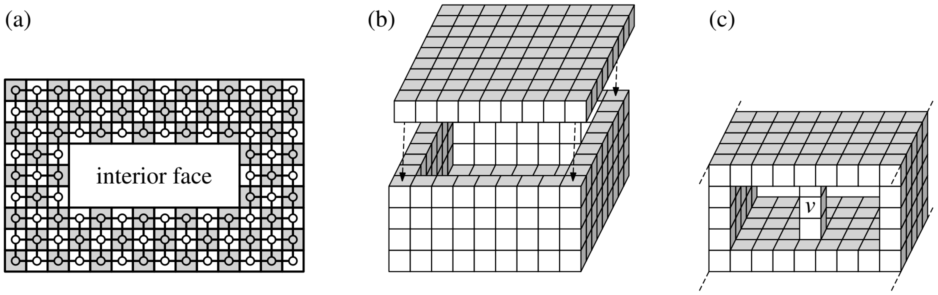

Fig. 8: Illustration for the hardness proof in 3D. (a) A 2-dimensional cage with thickness 3 and interior face of size 8×3. (b) A 3-dimensional cage with thickness 1 and interior of size $7 \times 3 \times 7$. (c) Attaching v to two sides of the box forces it into interior of cage.

by two parameters, the *thickness* t, which is an integer, and by the *interior* rectangle $w \times h$, with integer width and height. Given these parameters, the corresponding cage is the graph obtained from the $(2t+w) \times (2t+h)$ grid by deleting a $w \times h$ grid such that the distance from the external face to the large internal face corresponding to the interior is t. We call this internal face the *interior face*. Fig. 8a shows the cage with thickness 3 and interior 8×3 together with a contact representation with exactly one pixel per vertex.

Consider a pixel representation Γ of the cage of thickness t with interior $w \times h$. We show that either the bounding box of the interior face has size at most $w \times h$ or Γ uses at least one pixel per vertex plus t additional pixels. Thus, if we force some structure to lie in the interior of the cage, we can make the cost for using an area exceeding $w \times h$ arbitrarily large by increasing the thickness t appropriately.

We partition the cage into cycles $C_1, \dots, C_t$ where the vertices of C_i have distance i from the interior face. Consider C_1, which is the cycle bounding the interior face. The cycle C_1 has four *corner vertices* that are incident to two vertices in the outer face of C_1. All remaining vertices are incident to one vertex in the outer face. Requiring C_1 to be represented with exactly one pixel per vertex such that the corner vertices have two sides and every other vertex has one side incident to the outer face implies that C_1 must form a rectangle of size $(w+1) \times (h+1)$. Thus, if the bounding box of the interior face exceeds $w \times h$, C_1 requires at least one additional pixel. Moreover, the bounding box of the outer face of C_1 exceeds $(w+1) \times (h+1)$. Hence, an inductive argument shows that one requires at least one additional pixel for each cycle $C_1, \dots, C_t$, proving the claim.

Analogously, we can build cages in 3D with thickness t and interior $w \times h \times d$, by taking a 3D grid of size $(2t+w) \times (2t+h) \times (2t+d)$ and deleting a grid of size $w \times h \times d$. Fig. 8b shows the cage with $t = 1$, and $w \times h \times d = 7 \times 3 \times 7$. Assume that we have a graph G for which we want to find an minimum-size pixel representation (in 2D). We build a 3D cage, choose t, w and d to be very large, and set $h = 3$. To force G to lie in the interior of the cage, we pick a vertex v of G and connect it to two vertices of the cage as shown in Fig. 8c. This forces G to completely lie in the interior of the cage. As this interior has height 3 and no vertex of G (except for v) is allowed to touch another vertex of the cage, G is forced to lie in a single plane when choosing t sufficiently large (obviously, polynomial size is sufficient). Moreover, choosing w and d

sufficiently large, ensures that the size of the plane available for G does not restrict the possible representations of G. Finding a minimum-size pixel representation of G is equivalent to finding a minimum-size voxel representation of the resulting graph G'. $\square$