Convex Drawings of Graphs in Two and Three Dimensions (Preliminary Version)

Marek Chrobak* Michael T. Goodrich† Roberto Tamassia‡

Abstract

We provide $O(n)$-time algorithms for constructing the following types of drawings of n-vertex 3-connected planar graphs:
• 2D convex grid drawings with $(3n) \times (3n/2)$ area under the edge L_1-resolution rule;
• 2D strictly convex grid drawings with $O(n^3) \times O(n^3)$ area under the edge resolution rule;
• 2D strictly convex drawings with $O(1) \times O(n)$ area under the vertex-resolution rule, and with vertex coordinates represented by $O(n \log n)$-bit rational numbers;
• 3D convex drawings with $O(1) \times O(1) \times O(n)$ volume under the vertex-resolution rule, and with vertex coordinates represented by $O(n \log n)$-bit rational numbers.

We also show the following lower bounds:
• For infinitely many n-vertex graphs G, if G has a straight-line 2D convex drawing in a $w \times h$ grid satisfying the edge L_1-resolution rule then $w, h \geq 5n/6 + \Omega(1)$ and $w + h \geq 8n/3 + \Omega(1)$.
• For infinitely many bounded-degree triconnected planar graphs G with n vertices, any 3D convex drawing of G must have volume $2^{\Omega(n)}$ under the angular resolution rule.

1 Introduction and Overview

The research area of graph drawing is concerned with methods for automatically displaying a graph G so as to accent fundamental properties of G, while also optimizing important aesthetic qualities of the drawing, such as its size. It is a research area that combines computational geometry and graph theory to study interesting theoretical questions concerning algorithms for drawing graphs, as well as trade-offs for various geometric optimization criteria. Graph drawing algorithms have significant practical applications in computer graphics, software engineering, and databases.

In this paper, we investigate a classical geometric property in drawings of graphs: *convexity.* A *2D convex drawing* (see Fig. 1.a) is a planar straight-line drawing such that each face is a convex polygon. A *2D strictly convex drawing* (see Fig. 1.b) is a planar straight-line drawing such that each face is a strictly convex polygon. A *3D convex drawing* (see Fig. 1.c) is a realization of the graph by the skeleton of a 3D convex polytope. Convex drawings are important in visualization applications because of their aesthetic appeal. They have intrigued mathematicians for more than a century, with early work on the subject by Maxwell [40], Steinitz [47], and Tutte [50, 51].

Our work aims at characterizing the area/volume requirement of 2D/3D convex drawings. Of course, specifying a 2D area or a 3D volume bound begs the question of how this is to be measured, since one could reduce drawing dimensions by scaling. In order to prevent it, we will impose bounds on the minimum distances between vertices and (nonincident) edges. We define the following resolution measures:

vertex resolution: minimum distance between vertices;

edge resolution: minimum distance between an edge and a non-incident edge or vertex;

angular resolution: minimum angle between two edges incident at the same vertex.

In the above definition we assume that the distance is measured with the Euclidean metric L_2. For grid drawings it is convenient to use the L_1 metric instead of L_2. In that case, we will use the term *L_1-resolution* to indicate that we use L_1 metric.

With each of the above measures, we associate the corresponding *resolution rule.* We will restrict our attention in this paper to straight-line drawings that are drawn so as to achieve one of the following rules: The *vertex (edge) resolution rule* is that the vertex (edge) resolution is at least one. The

*Dept. of Computer Science, Univ. of California, Riverside, CA 92521, marek@cs.ucr.edu. This research supported by NSF under Grant CCR-9503498.

†Dept. of Computer Science, Johns Hopkins Univ., Baltimore, MD 21218, goodrich@cs.jhu.edu. This research supported by NSF under Grants IRI-9116843 and CCR-9300079, and by ARO under grant DAAH04-96-1-0013.

‡Dept. of Computer Science, Brown Univ., Providence, RI 02912, rt@cs.brown.edu. This research supported by NSF under Grant CCR-9423847 and by ARO under grant DAAH04-96-1-0013.

angular resolution rule states that the vertex resolution is verified, and that the angular resolution is at least $\alpha(d)$, where $\alpha(d)$ is a predefined function of the maximum degree of the graph.

The resolution rules make it possible to assign a meaningful measure to the area of the drawing. The three rules are motivated by the respective aesthetic desires that each vertex be distinguished from every other vertex, that each vertex be distinguished from each non-incident edge, and that each edge incident upon the same vertex be distinguished from its neighbors. Note that the vertex resolution rule is strictly weaker than requiring a *grid drawing* (integer coordinates for the vertices). However, the edge-resolution and angular-resolution rules can be either more or less restrictive than grid drawing requirement, depending on the drawing.

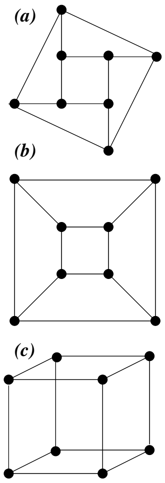

Figure 1: Convex drawings of a triconnected planar graph: (a) 2D convex drawing; (b) 2D strictly convex drawing; (c) 3D convex drawing.

1.1 Previous Related Work

In this section, we overview previous related work on drawings of graphs, with special attention to their area, volume, and convexity requirements. When measuring the area (volume) of a drawing, we consider the smallest axis-parallel box covering the drawing, and often use the notation $a \times b$ ($a \times b \times c$), referring to the length of the sides of the box. The area/volume requirement is affected by the type of resolution rule adopted for preventing vertices and edges to be placed arbitrarily close to one another.

2D Drawings. Straight line drawings of planar graphs are a classic topic in Mathematics, both in the plane [22, 46, 50, 51] and in 3-dimensions [26, 47]. Unfortunately, when translated into algorithms the proofs to these classic theorems produce drawings with poor resolution characteristics. Thus, recent attention has turned to area-efficient schemes for straight-line planar graph drawings, with the first breakthrough coming from de Fraysseix, Pach, and Pollack [14, 15], who show that any planar triangulation can be drawn as a straight line embedding in an $O(n) \times O(n)$ integer grid. Moreover, Chrobak and Payne [8] show that the approach of de Fraysseix *et al.* can be implemented in $O(n)$ time. Using a different and quite elegant approach, Schnyder [44] gives an alternate linear-time scheme for producing an $O(n) \times O(n)$ integer grid drawing of a triangulated planar graph, whose edge resolution is $O(1/n)$. Since then, several researchers have worked on extending and tightening these results in the integer grid model [6, 7, 34]. We will refer to the method from [14, 15, 8], as the *shift method*, as it works by successively adding vertices to the drawing and shifting horizontally parts of the existing drawing.

Several researchers have also considered tradeoffs involving the angular resolution (e.g., see [23, 24, 39]). For example, Garg and Tamassia [24] show that the problem of drawing a fixed-degree 3-connected planar graph under angular resolution in $\mathbf{R}^2$ requires exponential area. In addition, Di Battista, Tamassia, and Tollis [17] prove an interesting lower bound, which holds under any "reasonable" finite-resolution rule, that there exist an infinite family of planar acyclic digraphs such that for any digraph G in the class, any upward (i.e., with all edges "pointing up") planar straight-line drawing of G requires exponential area. Our formulation of the above resolution rules for 3D graph drawing extends these resolution notions.

2D Convex Drawings. Tutte [50, 51] shows that every triconnected planar graph admits a 2D strictly convex drawing, and that a 2D strictly convex drawing can be constructed by solving a certain sparse system of linear equations. Eades and Garvan [20] show that the drawings produced by Tutte's method have exponential area in the worst-case, under the vertex resolution rule.

Combinatorial characterizations of the graphs that admit 2D convex and strictly convex drawings are given by Tutte [50, 51], Thomassen [48, 49], Chiba, Yamanouchi, and Nishizeki [5], and Di Battista, Tamassia, and Vismara [18]. Linear time algorithms for constructing 2D convex drawings with real-valued coordinates are provided in [5]. This work is extended by Chiba, Onoguchi, and Nishizeki [4] to construct 2D "quasi convex" drawings for planar graphs that do not admit 2D convex drawings. Becker, Hotz and Osthof [2, 3] extend the notion of convex drawing to nonplanar graphs, and generalize some results of Tutte.

Kant [34, 33] presents a linear time algorithm for constructing 2D convex drawings with integer coordinates and $(2n-4) \times (n-2)$ area. Chrobak and Kant [6] and, independently, Schnyder and Trotter [45] reduce the grid size to $(n-2) \times (n-2)$. Lin and Skiena [35] (see also [1]) show that strictly convex drawings may require area $\Omega(n^3)$, since a strictly convex drawing of an n-vertex cycle requires such area. An on-line algorithm that tests whether a planar graph admits a 2D (strictly-) convex drawing in a dynamic environment where vertices and edges are incrementally inserted is given in [18].

3D Drawings. Due to the inherent "flat" nature of paper and most display hardware, it should come as no surprise that the vast majority of previous graph drawing research has focused on 2D drawings (e.g., see [16]). But recent advances in 3-dimensional visualization hardware have made 3D drawings technically feasible, and a handful of researchers (and film makers[1]) have begun to explore the possibilities of displaying graphs using this new technology [9, 13, 20, 21, 27, 32, 36, 42, 43].

3D Convex Drawings. The well-known Steinitz's theorem says that a graph admits a 3D convex drawing if and only if it is planar and triconnected [47] (see also Grünbaum [26]), properties that can be verified in linear time (see, e.g., [29, 30]). Interestingly, it is a easy exercise to derive from the published proofs of Steinitz's theorem a cubic-time method for constructing 3D convex drawings in the real-RAM model [41]. Unfortunately, this approach seems to require at least exponential volume and an exponential number of bits to implement.

Maxwell [40] (see also [10, 12, 52]) describes a mapping that transforms a 2D convex drawings with a certain "equilibrium property" into a 3D convex drawing. Further results on this transformation are given by Hopcroft and Kahn [31]. Eades and Garvan [20] show how to construct 3D convex drawings by combining the above transformation with the 2D-drawing method of Tutte [50, 51]. They also show that their drawings have exponential volume in the worst case. Smith (see [28]) claims a polynomial-time algorithm for constructing a 3D convex drawing inscribed in a sphere, with vertex coordinates represented by $O(n \log n)$-bit numbers, if a graph is known to be inscribable (which can be tested in linear time, e.g., for planar triangulations, due to a result of Dillencourt and Smith [19]). Das and Goodrich [13] present a linear-time algorithm for constructing a 3D convex drawing of a maximal planar graph such that the vertex coordinates are rational numbers that can be represented with a polynomial number of bits.

[1] An important plot element in the movie *Jurassic Park* involves a 3D virtual-reality traversal of a tree representing a Unix file system.

1.2 New Results

Let G be a triconnected planar graph with n vertices. We provide efficient algorithms for constructing the following types of drawings of G:

• a 2D convex grid drawing of G with $(3n) \times (3n/2)$ area under the edge L_1-resolution rule in linear time (previous methods achieved $\Omega(n^2 \times n^2)$ area);

• a 2D strictly convex grid drawing of G with $O(n^3) \times O(n^3)$ area under the edge resolution rule in linear time (it was not previously known how to achieve polynomial area);

• a 2D strictly convex drawing of G with $O(1) \times O(n)$ area under the vertex resolution rule, and with vertex coordinates represented by $O(n \log n)$-bit rational numbers in $O(n^{1.2})$ time (previous methods achieved $\Omega(n \times n)$ area);

• a 3D convex drawing of G with $O(1) \times O(1) \times O(n)$ volume under the vertex resolution rule, and with vertex coordinates represented by $O(n \log n)$-bit rational numbers in $O(n^{1.2})$ time (it was not previously known how to achieve polynomial volume).

We also show the following lower bounds on the area/volume of 2D/3D convex drawings under the edge/angular resolution rule:

• For infinitely many n-vertex graphs G, if G has a straight-line drawing in a $w \times h$ grid satisfying the edge L_1-resolution rule then $w, h \geq 5n/6 + \Omega(1)$ and $w + h \geq 8n/3 + \Omega(1)$ (previously it was known that $w, h \geq 2n/3$).

• For infinitely many bounded-degree triconnected planar graphs G with n vertices, any 3D convex drawing of G must have volume $2^{\Omega(n)}$ under the angular resolution rule (no nontrivial lower bound was previously known).

In the sections that follow we outline the main ideas behind each of the above results.

2 2D Convex Drawings

We begin with our results involving 2-dimensional convex drawings.

2.1 Improving Resolution for 2D Convex Drawings

Previous methods for straight-line drawings of planar graphs [8, 14, 15, 44] use grids of size $(2n-4) \times (n-2)$ or $(n-1) \times (n-1)$, and their vertex resolution is, obviously, at least one. However, their edge resolution for some graphs is only $O(1/n)$, and under the edge resolution rule they may require area as large as $\Omega(n^4)$.

In this section we show that further improvement of the aestheticity of straight-line drawings of planar graphs is possible, by providing a new grid drawing algorithm that uses a $(3n-7) \times (3n-7)/2$ grid, and thus only quadratic area, under the edge L_1-resolution rule. We find it interesting, that by increasing the grid size by a small constant factor, we can increase the edge resolution by an order of magnitude.

We use the concept of a canonical decomposition, as introduced by Kant [33, 34] which generalizes canonical orderings defined by de Fraysseix *et al.* [14, 15] for triangulated graphs.

Canonical Decompositions. Let G be an arbitrary, n-vertex, 3-connected plane graph and (v_1, v_2) an edge on the external face of G. Let $\pi = (V_1, \ldots, V_m)$ be a partition of V. By G_k we denote the subgraph of G induced by $V_k \cup \ldots \cup V_m$, and by C_k we denote the external face of G_k. We say that π is a *canonical decomposition of G with bottom edge* (v_1, v_2) if it satisfies the following conditions:

C.1. C_m is a face of G, and each C_k is a cycle containing (v_1, v_2).

C.2. Each G_k is 2-connected and internally 3-connected (that is, removing two internal vertices of G_k does not disconnect it).

C.3. For every $k = 2, \ldots, m-1$, one of the following conditions holds:

(a) $V_k = \{z\}$, for some z that belongs to C_k and has at least one neighbor in $G - G_k$.

(b) $V_k = (z_1, \ldots, z_\ell)$, where each z_i has at least one neighbor in $G - G_k$, z_1 and z_ℓ each have exactly one neighbor on C_{k+1}, and $z_2, \ldots, z_{\ell-1}$ have no neighbors in G_{k+1}.

If V_k satisfies Condition C.3.a, we call it a *singleton*; if it satisfies Condition C.3.b, we call it a *chain*. By 3-connectivity of G, V_1 must be a singleton. The following lemma was proven by Kant [33, 34]:

Lemma 2.1: *Every 3-connected plane graph has a canonical decomposition, and it can be constructed in linear time.*

Our algorithm, **ConvexDraw**, will add successively sets V_k in reverse order, adjusting the embedding at every step. By $f(v)$ we denote the current position of vertex v on the grid, i.e., $f(v) = (x(v), y(v))$. By $f(u,v)$ we denote the embedding of edge (u,v), that is, the line segment that connects $f(u)$ with $f(v)$. With each vertex w we will associate a set of vertices, $U(w)$, that contains vertices that have to be shifted right whenever w is shifted right. The set $U(w)$ changes during the execution of the algorithm. The general idea is that, unlike the previous approaches [14, 15, 8, 44], at the time when a new vertex is installed we shift all covered vertices to the right, ensuring that the are far from nonincident edges.

We give the details for **ConvexDraw** in the full version, proving the following theorem.

Theorem 2.2: *Algorithm ConvexDraw draws convexly every 3-connected planar graph in a $(3n-7) \times (3n-7)/2$ grid, under the edge L^1-resolution rule, and it can be implemented in linear time.*

Lower bound. What is the minimum grid size for grid drawings under the edge resolution rule? It is known that a grid of size $2n/3 \times 2n/3$ may be necessary for some graphs, even if there are no restrictions on edge resolution. We show the following lower bound (the proof will be given in the full version.)

Theorem 2.3: *For each $n \geq 1$ there exists a plane graph G_n on n vertices such that if G_n is embedded into a $w \times h$ grid under the edge L^1-resolution rule, then $h, w \geq 5n/6 + \Omega(1)$ and $h + w \geq 8n/3 + \Omega(1)$.*

2.2 Strictly Convex Drawings

In this section we consider *strictly convex* drawings of 3-connected planar graphs. We will show, using the results from the previous section, that a grid of size $O(n^3) \times O(n^3)$ is sufficient.

We assume we use a grid whose left-bottom corner is at $(0,0)$. If $f = (f_x, f_y)$ is a grid embedding of a 3-connected planar graph and D a positive integer, then Df is an enlargement of f with factor D, defined by $Df(u) = (Df_x(u), Df_y(u))$. If f is convex (not necessarily strictly) than a strictly convex drawing g is called *strictly convex D-adjustment* of f (or simply a D-adjustment) if $f(u) = Dg(u)$ for all vertices u at which f is strictly convex. Note that f must be strictly convex at three or more vertices on the external face.

Theorem 2.4: *Let f be a $3n \times 3n/2$ grid embedding of G produced by Algorithm ConvexGridDraw.*

Then there exists a D-adjustment of f for $D = cn^2$, if c is sufficiently large. Consequently, G has a strictly convex embedding into the $O(n^3) \times O(n^3)$ grid.

Proof: Let $D = cn^2$, for some c large enough. For simplicity, assume first that the external face of G is a triangle; we will deal with the general case later. Define a *straight segment* to be a maximum-length sequence of consecutive vertices on a face boundary such that all edges in-between form a straight line. Initially, we assume $g = Df$ and then we will perturbate vertices in the interiors of straight segments of the faces of G. Note that, by 3-connectivity, each vertex can belong to only one interior of a straight segment.

Algorithm ConvexDraw produces two types of straight segments: "bottom" segments, whose slope is either $45°$ or $-45°$, and "ceilings", which are horizontal segments on the top boundaries of faces. These boundaries of a face F are called *ceiling*(F) and *bottom*(F). Each vertex on F belongs to *ceiling*(F) or *bottom*(F). We also have side edges (left and right).

Pick an arbitrary straight segment $P = u_0 u_1 \ldots u_k$ with slope $45°$. Let $x_i = f_x(u_i)$ and define $\delta_i = (x_i - x_0)(x_k - x_i)$ for all i. For $i = 1, \ldots, k-1$, change the y-coordinates of u_i to $g_y(u_i) := Df_y(u_i) - \delta_i$. Other straight segments are perturbed in a similar way, always in the direction away from the face (vertically). Note that each vertex is shifted by at most $\delta = n^2/4$.

Since f satisfies the edge resolution rule, each vertex is at distance at least 1 from each edge. This distance will be $\geq cn^2$ in g. This implies the correctness of the embedding, since no vertex will cross any edge after perturbation. In the full version we show also that each face is strictly convex. ■

3 3D Convex Drawings

3.1 Stress Functions

Let G be a 3-connected planar graph embedded in $\mathbf{R}^2$. Such an embedding is *convex* if every face of G is convex. Let $(1, 2, \ldots, n)$ be a listing of the vertices of G and let $p_i = (x_i, y_i)$ denote the point in the plane corresponding to vertex i. A *stress* function defined on G is an assignment of weights $w_{i,j}$ so that $w_{i,j} = w_{j,i}$, for all $i \neq j$, and $w_{i,j} = 0$ if (i, j) is not an edge in G. A stress function is *convex* if the weight of each interior edge of G is (strictly) positive while the weight of each exterior edge is (strictly) negative. A stress function is merely *internally convex* if the weight of each interior edge is positive. A stress function w is at *equilibrium* for G if, for all i,

$$\sum_{j=1}^{n} w_{i,j}(p_i - p_j) = (0, 0) \qquad (1)$$

A stress function is at *internal equilibrium* if Equation (1) is guaranteed to hold only for the internal vertices of G. A stress function w' is an *external extension* of a function w if w' agrees with w on each internal edge of G. Tutte establishes an interesting connection between these properties of stress functions and the convexity of the embedding for G:

Theorem 3.1 [51]: *Let G be a 3-connected planar graph embedded in $\mathbf{R}^2$ to have a convex external face. If there exists an internally-convex stress function at internal equilibrium for G, then the embedding of G is convex.*

Tutte shows how to use this theorem to draw G. His approach is to embed convexly the external face of G, define $w_{i,j} = 1$ for each internal edge of G, and then solve the linear system determined by the boundary points and Equation (1) to determine the locations of all the internal vertices. Unfortunately, for our purposes, this approach does not in general produce nice drawings, for Eades and Garvan [20] show that such drawings can require exponential area under the vertex-resolution rule. Thus, if we are to achieve polynomial area using this approach, we must use a more "adaptive" approach. As a step in this direction we note the following useful result of Hopcroft and Kahn:

Lemma 3.2 [31]: *Let G be an embedded planar graph with triangular external face, and let w be an internally-convex stress at internal equilibrium for G. Then there is an external extension w' of w that is convex and at equilibrium for G.*

By Equation (1), an external extension w' can be computed from w in linear time simply by solving a linear system defined by the three external vertices (for there are only three undetermined variables).

3.2 3D Convex Drawings

There is a well-known duality between convex stress graphs and 3-dimensional convex polyhedra, dating back to Maxwell [40] (see also [10, 12, 52]). In this subsection we review the explicit formulation of Hopcroft and Kahn [31] for this mapping.

Let G be a convex embedding of a 3-connected planar graph and let G have a convex equilibrium stress w. With each face r in G associate a linear function $f_r(x, y) = a_r x + b_r y + c_r$. View G

as being embedded in the plane $z = 1$ and choose an arbitrary reference point $p_* = (x_*, y_*, 1)$ that is not collinear with any edge of G. The set of functions $\mathcal{F} = \{f_r\}$ defines a *w-consistent* mapping if, for each edge in G between points (p_i, p_j), incident upon faces r and s,

$$w_{i,j} = \frac{\delta(r,s)(f_s(x_*, y_*) - f_r(x_*, y_*))}{[p_i, p_j, p_*]}, \qquad (2)$$

where $[p_i, p_j, p_*] = \det([p_i, p_j, p_*])$ and $\delta(r, s)$ is the orientation coefficient, defined to be $+1$ if v_i precedes v_j in a counterclockwise ordering of the vertices around r, and -1 otherwise. Hopcroft and Kahn show that w-consistency is independent of the choice of reference point p_* (provided that it is not collinear with any edge of G).

Equation (2) may not by itself specify a unique w-consistent mapping $\mathcal{F}$. We may fix such an $\mathcal{F}$, however, by adding additional constraints implied by the topology, such as $f_r(x, y) = f_s(x, y)$ for any (x, y) on the line segment joining p_i and p_j. Given such $\mathcal{F}$, define a convex polyhedron by associating the plane $z = 1$ with the external face and the plane defined by f_r with each internal face r in G. Hopcroft and Kahn [31] show the following:

Theorem 3.3 [31]: *If w is a convex equilibrium stress for a convex embedding G, then the polyhedron defined by a w-consistent mapping is strictly convex.*

Thus, we have a template for producing 3-dimensional strictly-convex drawings of 3-connected planar graphs:

1. Construct an embedding of G with a convex equilibrium stress w.
2. Find a w-consistent mapping $\mathcal{F}$ to define a 3-dimensional convex polyhedron P that has G as its 1-skeleton.

This template forms a very high-level description of our approach, as well as that of Eades and Garvan [20]. Our algorithm differs from theirs significantly in Step 1, however.

Note that, under any of our resolution rules, if G has area A, then we can draw P to have volume A (by scaling the range of z-values to the interval $[0, 1]$). Let us therefore now concentrate on a method for drawing a 3-connected planar graph as a small-area planar convex equilibrium stress graph.

3.3 Computing a Convex Embedding with an x-Equilibrium Stress

Hopcroft and Kahn [31] show that there are convex planar embeddings that do not admit an equilibrium stress. Nevertheless, they show that embedded graphs that contain x-monotone spanning trees can be weighted to give a stress that satisfies Equation (1), for each $i \in \{1, 2, \ldots, n\}$, a condition we call *$x$-equilibrium.* Still, their method would not, in general, yield a convex stress. In this section we show that any 3-connected planar graph can be drawn as a small-area convex stress graph under the vertex resolution rule.

Let G be a 3-connected planar graph with a triangular external face (v_1, v_2, v_n). Suppose further that we are given a convex embedding of G in an $O(n) \times O(n)$ integer grid so that there are no vertical edges. This can be achieved by a simple modification of the 2-dimensional convex drawing algorithm of Chrobak and Kant [6], which we explore in the full version of this paper. Vertices v_1, v_2, v_n are mapped into the triangle with coordinates $(0, 0)$, $(4n, 0)$ and $(2n, 2n)$. Define the *x-cost*, $c_{i,j}$, of an edge (v_i, v_j) to be $|w_{i,j}(x_i - x_j)|$.

Lemma 3.4: *If G is an n-node 3-connected planar graph convexly embedded as above, then one can compute, in time $O(n)$, a convex x-equilibrium stress on G so that each x-cost $c_{i,j}$ is a positive integer with magnitude $O(n)$.*

Proof: Let us orient each edge in G from left to right (which is a well-defined notion, since G contains no vertical edges). Throughout this proof, (v_i, v_j) will denote an oriented edge, that is an edge of G such that $x_i < x_j$. By the assumptions of the lemma, for each internal edge (v_i, v_j), there exists a directed path P_{ij} from v_1 to v_n that contains (v_i, v_j). View the x-cost on each edge as a flow from left to right (with the x-equilibrium equation serving the role of flow conservation at each node). We do not set any capacity constraints on edges, however. The initial flow is 0 on all edges. Then for each (v_i, v_j), increase by 1 the flow along the path P_{ij} from v_1 to v_n. Since we maintain internal x-equilibrium with each "augmentation," this procedure will result in an internally-convex stress function that is at internal x-equilibrium. This can be extended to a convex stress at x-equilibrium by Lemma 3.2. Moreover, the flow on any internal edge is increased by 1 at most $3n$ times; hence, the x-cost on any internal edge is at most $3n$. By the proof of Lemma 3.2, this implies that all x-costs in G are integers bounded by $O(n)$.

The above method works in time $O(n^2)$. In order to achieve a running time of $O(n)$, we carefully pick the augmenting paths P_{ij}. At each vertex pick one incoming edge. This defines a tree T_1 rooted at v_1. Symmetrically define tree T_2 rooted at v_n by picking one outgoing edge from each vertex. Define P_{ij} as the concatenation of the path from v_1 to v_i in T_1 (that we call the prefix of P_{ij}), edge (v_i, v_j), and the path from v_j to v_n in T_2 (called the suffix of P_{ij}).

Recall that the flow c_{ab} on an edge (v_a, v_b) is the number of augmenting paths P_{ij} that contain this edge, which can be expressed as $c_{ab} = 1 + p_{ab} + s_{ab}$, where p_{ab} and s_{ab} are, respectively, the numbers of prefixes and suffixes of the augmenting paths containing (v_a, v_b). We have $p_{ab} = 0$ if $(v_a, v_b) \notin T_1$. To compute p_{ab} for edges $(v_a, v_b) \in T_1$, we traverse T_1 in postorder. When backtracking from v_b to v_a, we set $p_{ab} = \sum_{(v_b,v_d)\in T_s} p_{bd} + \sum_{(v_b,v_d)} 1$. The numbers s_{ab} are computed similarly using T_2. ■

Thus, we can take the above convex embedding of a 3-connected planar graph G and in time $O(n)$ produce a convex x-equilibrium stress for G. This stress function will in general not be at y-equilibrium, however.

3.4 Computing a Convex Embedding with an Equilibrium Stress

Nevertheless, we can easily convert such a drawing into a convex equilibrium stress graph. In particular, we let $Ax = b$ denote the linear system defined by the weight function, which achieves x-equilibrium, Equation (1), and the boundary conditions fixing the exterior triangle for G. Since all the equations in this system involving x-coordinates are already satisfied, solving the system $Ax = b$ finds the y-coordinates of the vertices of G that produce a convex equilibrium stress graph embedding G' for G, while keeping the x-coordinates unchanged.

This algorithm clearly produces a convex embedding of G in the plane together with a convex equilibrium stress defined on this embedding, by Theorem 3.1. Moreover, if we start with G being embedded in an $O(n) \times O(n)$ integer grid, then G' will be a convex embedding such that each x-coordinate is a positive integer with magnitude $O(n)$, and G' will have no vertical edges. In addition, by well-known properties of rational-arithmetic linear system solving, we can guarantee that the number of bits needed to represent any y-coordinate, as a rational number, is $O(n \log n)$. If we scale the y-coordinates to lie in the interval $[0, 1]$, then the drawing will still be a convex equilibrium stress embedding, but will have area $O(n)$ under the vertex-resolution rule. Thus, we have the following:

Theorem 3.5: *Given a 3-connected planar graph G, one can produce a convex equilibrium stress embedding of G with $O(n)$ area under the vertex-resolution rule. The running time needed to achieve this is $O(P(n))$, where $P(n)$ is the time needed to solve an $n \times n$ linear system defined by planar constraints.*

Note that this area bound contrasts sharply with the exponential lower bound of Eades and Garvan [20] for the area of Tutte drawings under the vertex-resolution rule.

Incidentally, there are fairly simple separator-based methods [25, 37, 38] for achieving an $O(n^{1.5})$ bound for $P(n)$, while much more sophisticated methods allow one to achieve an $O(M(n^{1/2}))$ bound, where $M(n)$ is the time needed to multiply two $n \times n$ matrices (the current best bound for $M(n)$ is $O(n^{2.375})$ [11]). Thus, by our template, we have the following:

Theorem 3.6: *Given a 3-connected planar graph G, in time $O(M(n^{1/2}))$ one can draw G as a convex polyhedron in $\mathbf{R}^3$ using $O(n)$ volume under the vertex-resolution rule.*

Thus, under current theoretical definition of $M(n)$ [11], we can achieve a running time of $O(n^{1.19})$, but in practice the $O(n^{1.5})$ bound is probably more realistic.

3.5 On Angular Resolution and Volume of 3D Drawings

In this section we show that under the angular resolution rule there are 3-connected planar graphs that require exponential volume to draw as 3-dimensional convex polyhedra. We establish this lower bound via a reduction from the problem of drawing a fixed-degree 3-connected planar graph under angular resolution in $\mathbf{R}^2$, which was shown to require exponential area by Garg and Tamassia [24].

The main difficulty in extending their proof to convex drawings in $\mathbf{R}^3$ is that the third dimension allows a tremendous amount of extra drawing freedom. For example, a convex drawing in $\mathbf{R}^3$ can achieve angular resolution and yet have many 2-dimensional projections that do not achieve angular resolution. The main idea of our lower bound construction is to demonstrate an n-node 3-connected planar graph G_n such that any 3D convex drawing of G_n that achieves angular resolution contains a connected subgraph of size $\Theta(n)$ that projects to a 2D drawing that also achieves angular resolution. By the lower bound of Garg and Tamassia [24], this would establish an exponential lower bound on the area of this projection, hence the volume of this drawing would also be at least exponential.

We define G_n algorithmically. We begin with a 17-node cycle P_{17}, which will form a face in G_n, hence P_{17} must be drawn in some plane in $\mathbf{R}^3$. So, let P'_{17} be a planar drawing of P_{17} as a convex polygon. Orient each edge of P'_{17} in the clockwise direction. For a vertex v on P'_{17}, let $p(v)$ and $s(v)$ respectively denote the predecessor edge and successor edge incident upon v in this orientation. Define the *external angle* $\beta(v)$ at v to be the angle formed at v between an extension of $p(v)$ (as a ray with $p(v)$'s orientation) and an extension of $s(v)$. Also, following Grünbaum [26], let us measure angles as fractions of 1 (so that a right angle is 1/4).

Lemma 3.7: *P'_{17} has two consecutive vertices with external angles less than* $1/8$.

Proof: In a convex polygon P_n we have $\sum_{v \in P_n} \beta(v) = 1$. $P_{17'}$ can have at most 8 vertices with external angle at least 1/8. Thus, P'_{17} must have at least 9 vertices with external angle less than 1/8. Moreover, by a simple pigeon-hole argument, two of these vertices must be consecutive. ■

Let us continue, then, with our definition of G_n. Our next augmentation is to add a vertex v^* that is adjacent to each vertex on P_{17} (so as to define a pyramid). Let Q denote this new graph. For each edge e of Q incident upon v^* define the *external angle*, $\beta(e)$, at e analogously to the planar external angle at a vertex. Specifically, define $\beta(e)$ to be the fraction of the sphere defined between the two planes incident upon e and oriented in a clockwise direction. Define an edge e to be *shallow* if $\beta(e) \leq 1/8$. By Lemma 3.7, we know that, no matter where v^* is placed, two consecutive edges incident upon v^* must be shallow.

We wish to force there to be a triangle τ in Q with all three of its edges being shallow. This is because any subgraph placed in the interior of τ and drawn to achieve angular resolution would project to the plane containing τ so as to achieve (2-dimensional) angular resolution. This would then allow us to complete the proof by placing the graph, H_k, used in the 2-dimensional lower bound of Garg and Tamassia [24], in the interior of τ. Let us therefore augment Q with additional triangular faces in a fashion that will allow us to argue that there must be at least one triangular face with three shallow edges. If we can accomplish this by adding just a constant number of additional edges to Q, then we can place G_k in the interior of each such face to complete the proof.

Let t be the triangular face of Q with two shallow edges. If t actually has three shallow edges, then we are done, so let us assume that the third edge of t is not shallow. Of course, it must nevertheless have measure less than 1/2. Define the *stellation* of a triangular face s to be the placement of a new vertex in the interior of s which is then made to be adjacent to the three vertices of s. We start with t and stellate it. This creates two triangular faces t_1 and t_2 that are incident upon v^* and a triangular face that is not incident upon v^*. Let us therefore repeat this procedure, likewise stellating t_1 and t_2. This creates four new triangular faces incident upon v^* and two new edges incident upon v^* as well. Let us continue to iterate this procedure, stellating all the triangular faces incident upon v^* in each iteration. We repeat this procedure for a total of ℓ iterations, thus obtaining a subgraph S_t. It is useful to note that the planar dual of S_t is a depth-ℓ complete binary tree B with additional edges connecting the leaves of B. We can show that there exists a sufficiently large integer constant ℓ such that at least one triangle τ of S_t has three shallow edges.

To sum up, then, our construction of G_n starts with P_{17}, adds v^* to be adjacent to each vertex of P_{17}, augments each triangle incident to v^* to become the subgraph S_t, and then adds the lower-bound graph H_k of Garg and Tamassia [24] in the interior of each triangle in a S_t to complete the proof. If the resulting graph, G_n, is drawn as a convex polyhedron in $\mathbf{R}^3$ so as to achieve angular resolution, then, by the above argument, at least one of these H_k's will project to a plane so as to preserve angular resolution. But by the lower bound of Garg and Tamassia, such a projection must have area at least $2^{\Omega(n)}$; hence, the drawing in $\mathbf{R}^3$ must have volume at least $2^{\Omega(n)}$. We conclude:

Theorem 3.8: *There is a fixed-degree n-node 3-connected planar graph G_n that requires $2^{\Omega}(n)$ volume to draw as a convex polyhedron in $\mathbf{R}^3$ under the angular resolution rule, with $\alpha(v) > \alpha_0$ for any fixed constant $\alpha_0 > 0$.*

Acknowledgements. We would like to thank Robert Cohen, Peter Eades, and Patrick Garvan for helpful discussions and e-mail exchanges regarding topics related to this paper.

References

[1] G. E. Andrews. A lower bound for the volume of strictly convex bodies with many boundary lattice points. *Trans. Amer. Math. Soc.*, 106:270–279, 1963.

[2] B. Becker and G. Hotz. On the optimal layout of planar graphs with fixed boundary. *SIAM J. Comput.*, 16(5):946–972, 1987.

[3] B. Becker and H. G. Osthof. Layout with wires of balanced length. *Inform. Comput.*, 73:45–58, 1987.

[4] N. Chiba, K. Onoguchi, and T. Nishizeki. Drawing planar graphs nicely. *Acta Inform.*, 22:187–201, 1985.

[5] N. Chiba, T. Yamanouchi, and T. Nishizeki. Linear algorithms for convex drawings of planar graphs. In J. A. Bondy and U. S. R. Murty, editors, *Progress in Graph Theory*, pages 153–173. Academic Press, New York, NY, 1984.

[6] M. Chrobak and G. Kant. Convex grid drawings of 3-connected planar graphs. Technical Report RUU-93-45, Dept. of Computer Sci., Utrecht Univ., 1993.

[7] M. Chrobak and S. Nakano. Minimum-width grid drawings of plane graphs. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 104–110. Springer-Verlag, 1995.

[8] M. Chrobak and T. Payne. A linear-time algorithm for drawing planar graphs. *Information Processing Letters*, 54:241–246, 1995.

[9] R. F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 1–11. Springer-Verlag, 1995.

[10] R. Connelly. Rigidity and energy. *Invent. Math.*, 66:11–33, 1982.

[11] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Computing*, 9:251–280, 1990.

[12] H. Crapo and W. Whitely. Statics of frameworks and motions of panel structures, a projective geometric introduction. *Structural Topology*, 6:42–82, 1982.

[13] G. Das and M. T. Goodrich. On the complexity of approximating and illuminating three-dimensional convex polyhedra. In *Proc. 4th Workshop Algorithms Data Struct.*, volume ?? of *Lecture Notes in Computer Science*, page ?? Springer-Verlag, 1995.

[14] H. de Fraysseix, J. Pach, and R. Pollack. Small sets supporting Fary embeddings of planar graphs. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 426–433, 1988.

[15] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.

[16] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4:235–282, 1994.

[17] G. Di Battista, R. Tamassia, and I. G. Tollis. Area requirement and symmetry display of planar upward drawings. *Discrete Comput. Geom.*, 7:381–401, 1992.

[18] G. Di Battista, R. Tamassia, and L. Vismara. On-line convex planarity testing. In *Graph-Theoretic Concepts in Computer Science (Proc. WG 94)*, Lecture Notes in Computer Science. Springer-Verlag, 1994.

[19] M. B. Dillencourt and W. D. Smith. A linear-time algorithm for testing the inscribability of trivalent polyhedra. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 177–185, 1992.

[20] P. Eades and P. Garvan. Drawing stressed planar graphs in three dimensions. In *Graph Drawing '95*, Lecture Notes in Computer Science. Springer-Verlag, 1995.

[21] P. Eades, C. Stirk, and S. Whitesides. The techniques of Komolgorov and Bardzin for three dimensional orthogonal graph drawings. Manuscript, Dept. of Computer Sci., Univ. of Newcastle, 1995.

[22] I. Fary. On straight lines representation of planar graphs. *Acta Sci. Math. Szeged.*, 11:229–233, 1948.

[23] M. Formann, T. Hagerup, J. Haralambides, M. Kaufmann, F. T. Leighton, A. Simvonis, E. Welzl, and G. Woeginger. Drawing graphs in the plane with high resolution. *SIAM J. Comput.*, 22:1035–1052, 1993.

[24] A. Garg and R. Tamassia. Planar drawings and angular resolution: Algorithms and bounds. In *Proc. 2nd Annu. European Sympos. Algorithms (ESA '94)*, volume 855 of *Lecture Notes in Computer Science*, pages 12–23. Springer-Verlag, 1994.

[25] J. R. Gilbert and R. E. Tarjan. The analysis of a nested dissection algorithm. *Numer. Math.*, 50:377–404, 1987.

[26] B. Grünbaum. *Convex Polytopes.* Wiley, New York, NY, 1967.

[27] S. M. Hashemi and I. Rival. Upward drawings to fit surfaces. In *Order, Algorithms, and Applications (Proc. ORDAL '94)*, volume 831 of *Lecture Notes in Computer Science*, pages 53–58. Springer-Verlag, 1994.

[28] C. D. Hodgson, I. Rivin, and W. D. Smith. A characterization of convex hyperbolic polyhedra and of convex polyhedra inscribed in the sphere. *Bull. (New Series) of the AMS*, 27(2):246–251, 1992.

[29] J. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2:135–158, 1973.

[30] J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[31] J. E. Hopcroft and P. J. Kahn. A paradigm for robust geometric algorithms. *Algorithmica*, 7:339–380, 1992.

[32] T. Jéron and C. Jard. 3D layout of reachability graphs of communicating processes. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 25–32. Springer-Verlag, 1995.

[33] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica* (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia). (to appear).

[34] G. Kant. Drawing planar graphs using the *lmc*-ordering. In *Proc. 33th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 101–110, 1992.

[35] Y. Lin and S. Skiena. Complexity aspects of visibility graphs. Technical Report 92/08, SUNY Stony Brook, 1992.

[36] G. Liotta and G. Di Battista. Computing proximity drawings of trees in the 3-dimemsional space. In *Proc. 4th Workshop Algorithms Data Struct.*, volume ?? of *Lecture Notes in Computer Science*, page ?? Springer-Verlag, 1995.

[37] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM J. Numer. Anal.*, 16(2):346–358, 1979.

[38] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9:615–627, 1980.

[39] S. Malitz and A. Papakostas. On the angular resolution of planar graphs. *SIAM J. Discrete Math.*, 7:172–183, 1994.

[40] J. C. Maxwell. On reciprocal figures and diagrams of forces. *Phil. Mag. Ser.*, 27:250–261, 1864.

[41] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction*. Springer-Verlag, New York, NY, 1985.

[42] S. P. Reiss. 3-D visualization of program information. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, pages 12–24. Springer-Verlag, 1995.

[43] S. P. Reiss. An engine for the 3D visualization of program information. *J. Visual Languages and Computing* (special issue on Graph Visualization, edited by I. F. Cruz and P. Eades), 6(3), 1995.

[44] W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 138–148, 1990.

[45] W. Schnyder and W. Trotter. Convex drawings of planar graphs. *Abstracts of the AMS*, 92T-05-135, 1992.

[46] S. K. Stein. Convex maps. *Proc. Amer. Math. Soc.*, 2:464–466, 1951.

[47] E. Steinitz and H. Rademacher. *Vorlesungen über die Theorie der Polyeder*. Julius Springer, Berlin, Germany, 1934.

[48] C. Thomassen. Planarity and duality of finite and infinite planar graphs. *J. Combin. Theory Ser. B*, 29:244–271, 1980.

[49] C. Thomassen. Plane representations of graphs. In J. A. Bondy and U. S. R. Murty, editors, *Progress in Graph Theory*, pages 43–69. Academic Press, New York, NY, 1984.

[50] W. T. Tutte. Convex representations of graphs. *Proceedings London Mathematical Society*, 10:304–320, 1960.

[51] W. T. Tutte. How to draw a graph. *Proceedings London Mathematical Society*, 3(13):743–768, 1963.

[52] W. Whitney. Motions and stresses of projected polyhedra. *Structural Topology*, 7:13–38, 1982.