Simultaneous Embedding of a Planar Graph and Its Dual on the Grid*

Cesim Erten and Stephen G. Kobourov

Department of Computer Science University of Arizona
{cesim,kobourov}@cs.arizona.edu

Abstract. Traditional representations of graphs and their duals suggest the requirement that the dual vertices should be placed inside their corresponding primal faces, and the edges of the dual graph should cross only their corresponding primal edges. We consider the problem of simultaneously embedding a planar graph and its dual on a small integer grid such that the edges are drawn as straight-line segments and the only crossings are between primal-dual pairs of edges. We provide an $O(n)$ time algorithm that simultaneously embeds a 3-connected planar graph and its dual on a $(2n-2) \times (2n-2)$ integer grid, where n is the total number of vertices in the graph and its dual.

Key Words. Graph drawing, planar embedding, simultaneous embedding, convex planar drawing.

1 Introduction

In this paper we address the problem of simultaneously drawing a planar graph and its dual on a small integer grid. The *planar dual* of an embedded planar graph G is the graph G' formed by placing a vertex inside each face of G, and connecting those vertices of G' whose corresponding faces in G share an edge. Each vertex in G' has a corresponding primal face and each edge in G' has a corresponding primal edge in the original graph G. The traditional manual representations of a graph and its dual, suggest two natural requirements. One requirement is that we place a dual vertex inside its corresponding primal face and the other is that we draw a dual edge so that it only crosses its corresponding primal edge. We provide a linear-time algorithm that simultaneously draws a planar graph and its dual using straight-line segments on the integer grid while satisfying these two requirements.

1.1 Related Work

Straight-line embedding a planar graph G on the grid, i.e., mapping the vertices of G on a small integer grid such that each edge can be drawn as a straight-line segment and that no crossings between edges are created, is a well-studied graph drawing problem. The first solution to this problem was given by de Fraysseix,

* A full version of this extended abstract is at www.cs.arizona.edu/~cesim/dual.ps.

Pach and Pollack [6] who provide an algorithm that embeds a planar graph on n vertices on the $(2n-4) \times (n-2)$ integer grid. Later, Schnyder [13] present another method that requires grid size $(n-2) \times (n-2)$. Also, several restrictions of this problem have been considered. Harel and Sardas [7] provide an algorithm to embed a biconnected graph on the $(2n-4) \times (n-2)$ grid without triangulating the graph initially. The algorithm of Chrobak and Kant [4] embeds a 3-connected planar graph on a $(n-2) \times (n-2)$ grid so that each face is convex. Miura, Nakano, and Nishizeki [11] further restrict the graphs under consideration to 4-connected planar graphs with at least 4 vertices on the outer face and present an algorithm for straight-line embedding of such graphs on a $(\lceil n/2 \rceil - 1) \times (\lfloor n/2 \rfloor)$ grid.

In a paper dating back to 1963, Tutte [14] shows that there exists a simultaneous straight-line representation of any planar graph and its dual in which the only intersections are between corresponding primal-dual edges. However, a disadvantage of this representation is that the area required by the algorithm can be exponential in the number of vertices of the graph.

Brightwell and Scheinerman [2] show that every 3-connected planar graph G can be represented as a collection of circles, one circle representing each vertex and each face, so that, for each edge of G, the four circles representing the two endpoints and the two neighboring faces meet at a point. Moreover, the vertex-circles cross the face-circles at right angles. This result implies that one can represent a 3-connected planar graph and its dual simultaneously in the plane with straight-line edges so that the primal edges cross the dual edges at right angles (provided that the vertex corresponding to the unbounded face is located at infinity). Mohar [12] extends the results of [2] by presenting an approximation algorithm that given a 3-connected planar graph $G = (V, E)$ and a rational number $\epsilon > 0$ finds an ϵ-approximation for the radii and the coordinates of the centers for the primal-dual circle representation for G and its dual. Mohar's algorithm runs in time polynomial in $|E(G)|$ and $\log(1/\epsilon)$ and the angles of the primal-dual edge crossings are arbitrarily close to $\pi/2$.

Bern and Gilbert [1] address a variation of the simultaneous planar-dual embedding problem: finding suitable locations for dual vertices, given a straight-line planar embedding of a planar graph, so that the edges of the dual graph are also straight-line segments and cross only their corresponding primal edges. They present a linear time algorithm for the problem in the case of convex 4-sided faces and show that the problem is NP-hard for the case of convex 5-sided faces.

1.2 Our Results

The simultaneous embedding in [2] guarantees right angles for the primal-dual edge crossings where the unbounded face needs to be handled in a special way by creating a vertex at infinity. Even without considering the unbounded face, the methods in [2] and [12] do not provide bounds on the area required for the simultaneous embedding and they are less practical than our approach.

In this paper we present an algorithm for embedding a given planar graph G and its dual simultaneously so that following conditions are met:

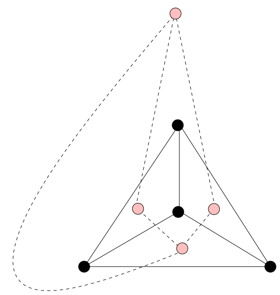

Fig. 1. A drawing of K_4 and its dual. If the vertex corresponding to the outer face is drawn explicitly, then one of its adjacent edges must have a bend.

- The primal graph is drawn with straight-line segments without crossings.
- The dual graph is drawn with straight-line segments without crossings.
- Each dual vertex lies inside its primal face.
- A pair of edges cross if and only if the edges are a primal-dual pair.
- Both the primal and the dual vertices are on the $(2n-2) \times (2n-2)$ grid, where n is the number of vertices in the primal and dual graphs.
- The running time of the algorithm is $O(n)$.

Similar to most primal-dual representation methods, the unbounded (outer) face must be treated differently. If the vertex corresponding to the unbounded face is not explicitly drawn in the plane, then all of the conditions above are met. However, if it is drawn explicitly, then one of the dual edges emanating from it cannot be a straight-line segment; see Fig. 1. In our grid-embedding algorithm, we provide an option for not drawing the vertex representing the outer face explicitly, or if it is drawn, then one of the edges emanating from it has one bend (that also is on the grid). Note, that if the embedding is done on the surface of a sphere, the edges emanating from this vertex are arcs of great circles and the unbounded face does not require special treatment.

In section 2 we describe the algorithm in detail and in section 3, we briefly discuss the implementation and present several drawings of primal-dual graphs produced by our algorithm.

2 Algorithm for Embedding a Graph and Its Dual

Let G_1 be a 3-connected planar graph. We construct a new graph G_2 that combines information about both the planar graph G_1 and its dual. For this construction we make some changes in G_1. We introduce a new vertex ${v_i}'$ corresponding to a face ${\mathcal{F}_i}'$ of G_1, for all $1 \leq i \leq f$, where f is the number of faces of G_1. We connect each newly added vertex ${v_i}'$ to each vertex v_j of ${\mathcal{F}_i}'$ with a single new edge and delete all the edges that originally belonged to G_1. Fig. 2 shows a sample construction. We call the resulting planar graph G_2 *fully-quadrilateralated (FQ)*, i.e., every face of G_2 is a quadrilateral. Since the original graph G_1 is 3-connected, the resulting graph G_2 is also 3-connected (proven formally in [14]).

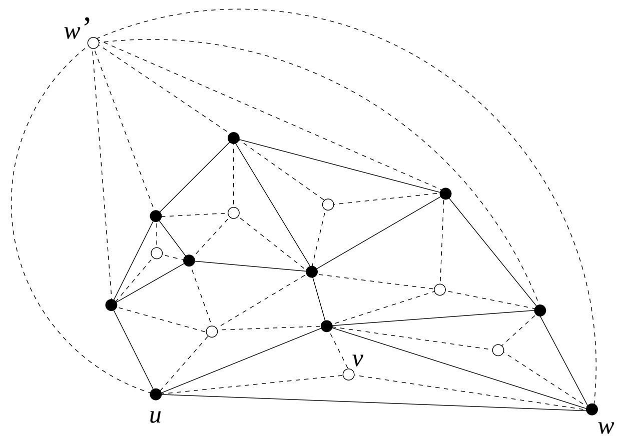

Fig. 2. Creating graph G_2: the original 3-connected graph G_1 is drawn with solid lines and filled-in circles; we insert the dual vertices (drawn as empty circles) and add the edges connecting primal and dual vertices (drawn as dashed lines). To obtain G_2 we remove the original edges of G_1 (drawn with solid lines).

Observation: If we can embed the graph G_2 on the grid so that each inner face of G_2 is strictly convex and the outer face of G_2 lies on a strictly concave quadrilateral, then we can embed the initial graph G_1 and its dual so that we meet all the problem requirements with the only exception that one edge of the primal graph G_1(or its dual) is drawn with one bend.

The requirement that the edges of the dual graph be straight and cross only their corresponding primal edges is guaranteed by the strict convexity of the quadrilateral faces. Let the outer face of the graph G_2 be (u, v, w, w'), where u, w are primal vertices and v, w' are dual vertices, as shown in Fig. 2. The exception arises from the fact that we need to draw (u, w) and (v, w'), while both of these edges can not lie inside the quadrilateral (u, v, w, w'). In order to get around this problem we embed the quadrilateral (u, v, w, w') so that it is strictly concave. This way only one bend for one of the edges (u, w) or (v, w') will be sufficient. As a result all the edges in the primal and the dual graph are straight-line edges, except for one edge. In fact, it is easy to choose the exact edge we need (either from the primal or from the dual).

Hence, the original problem can be transformed into a problem of straight-line embedding an FQ-3-connected planar graph G on the grid so that each internal face of G is strictly convex and the outer face of G lies on a strictly concave quadrilateral. Note that this problem can be solved by the algorithm of Chrobak *et al.* [3]. However, the area guaranteed by their algorithm is $O(n^3) \times O(n^3)$, whereas our algorithm guarantees a drawing on the $(2n-2) \times (2n-2)$ grid, which is stated in the main theorem in this paper:

Theorem 1. *Given a 3-connected planar graph G_1, we can embed G_1 and its dual on a $(2n-2)\times(2n-2)$ grid, where n is the number of vertices in G_1 and its dual, so that each dual vertex lies inside its primal face, each dual edge crosses* *only its primal edge and every edge in the overall embedding is a straight-line segment except for one edge which has a bend placed on the grid. Furthermore, the running time of the algorithm is $O(n)$.*

2.1 Overview of the Algorithm

Given a 3-connected graph G_1, we summarize our algorithm to simultaneously embed G_1 and its dual as follows:

- Find a topological embedding of G_1 using [8].
- Apply the construction described above to find G_2.
- Let $G = G_2$, where G is an FQ-3-connected planar graph.
- Find a suitable canonical labeling of the vertices of G.
- Place the vertices of G on the grid one at a time using this ordering.
- Remove all the edges of G and draw the edges of G_1 and its dual.

Note that our method works only for 3-connected graphs. A commonly used technique for drawing a general planar graph is to embed the graph after fully triangulating it by adding some extra edges and then to remove the extra edges from the final embedding. Using the same idea, we could first fully triangulate any given planar graph. Then after embedding the resulting 3-connected planar graph and its dual, we could remove the extra edges that were inserted initially. However, the problem with this approach is that after removing the extra edges there could be faces with multiple dual vertices inside. Thus the issue of choosing a suitable location for the duals of such faces remains unresolved. In fact, depending on the drawing of that face, it could as well be the case that no suitable location for the dual exists [1]. In the rest of the paper we consider only 3-connected graphs.

2.2 The Canonical Labeling

We present the canonical labeling for the type of graphs under consideration. It is a simple restriction of the canonical labeling of [9], which in turn is based on the ordering defined in [6].

Let G be an FQ-3-connected planar graph with n vertices. Let (u, v, w, w') be the outer face of G s.t. u, w are primal vertices and v, w' are dual vertices. Then there exists a mapping δ from the vertices of G onto v_i, $1 \leq i \leq m$ such that δ maps u and v to v_1, w' to v_m and satisfies the following invariants for every $3 \leq k \leq m$:

1. The subgraph $G_{k-1} \subseteq G$, induced by the vertices labeled v_i, $1 \leq i \leq k-1$ is biconnected and the boundary of its exterior face is a cycle C_{k-1} containing the edge (u, v).
2. Either one vertex or two vertices can be labeled v_k.
   (a) Let z_0 be the only vertex labeled v_k. Then z_0 belongs to the exterior face of G_{k-1}, has at least two neighbors in G_{k-1} and at least one neighbor in $G - G_k$.

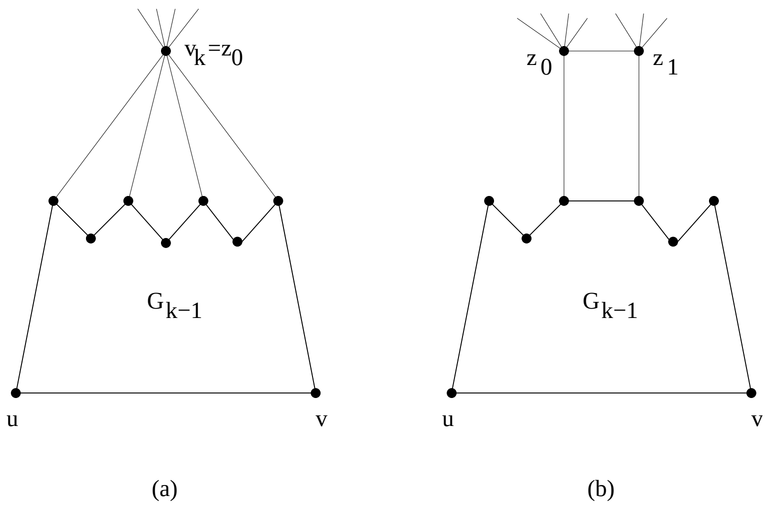

Fig. 3. (a) One vertex, z_0, is labeled v_k; (b) Two vertices, z_0 and z_1, are labeled v_k.

(b) Let z_0, z_1 be the two vertices labeled v_k, where (z_0, z_1) is an edge in G. Then z_0, z_1 belong to the outer face of G_{k-1}, each has exactly one neighbor in G_{k-1} and at least one neighbor in $G - G_k$.

Since G is FQ, all the faces created by adding v_k, $3 \leq k \leq m$, have to be quadrilaterals; see Fig. 3.

Note that assigning the mappings onto v_1 and v_m as above provides us the embedding where all the edges of both the primal and the dual graph are straight except for one primal edge, (u, w), which has a bend. Alternatively assigning v and w to map onto v_1, and u to map onto v_m would choose a dual edge, (v, w'), to have a bend.

Lemma 1. *Every FQ-3-connected planar graph has a canonical labeling as defined above.*

Kant [9] provides a linear-time algorithm to find a canonical labeling of a general 3-connected planar graph. It is easy to see that the canonical labeling definition of [9] when applied to FQ-3-connected planar graphs, gives us the labeling defined above.

2.3 The Placement of the Vertices

The main idea behind most of the straight-line grid embedding algorithms is to come up with a suitable ordering of the vertices and then place the vertices one at a time using the given order, while making sure that the newly placed vertex (or vertices) is (are) visible to all the neighbors. In order to realize this last goal, at each step, a set of vertices are shifted to the right without affecting the planarity of the drawing so far. Our placement algorithm is similar to the algorithm of Chrobak and Kant [4], with some changes in the invariants that we maintain to guarantee the visibility together with strict convexity of the faces.

Let the canonical labeling, δ, that maps the vertices of G onto $v_1, v_2, ...v_m$ be defined as in the previous section. Let $\mathcal{U}(g_i)$ denote the vertices *under* g_i. $\mathcal{U}(g_i)$ should be shifted to the right whenever the vertex g_i is shifted to the right. $\mathcal{U}(g_i)$ is initialized to $\{g_i\}$ for every vertex g_i of G. Let $\delta(g_i) = v_{i'}$ and $\delta(g_j) = v_{j'}$. Then we define $Low(g_i, g_j) = i$ if $i' < j'$, $Low(g_i, g_j) = j$ if $j' < i'$. If $i' = j'$ then let $Low(g_i, g_j)$ be the one that is placed to the left. Let $x(g_i)$, $y(g_i)$ respectively denote the x and y coordinates of the vertex g_i.

•*Embed the First Quadrilateral Face:* We begin by placing the vertices mapped onto v_1 and v_2. The ones that are mapped onto v_1 are u and v. We place u at $(0, 0)$ and v at $(3, 0)$. Note that two vertices should be mapped to v_2. We place the vertex that is mapped to v_2 and that has an edge with u at $(1, 1)$ and the other at $(2, 1)$.

Then, for every k, $3 \leq k \leq m$, we do the following:

•*Update $\mathcal{U}(g_i)$:* Let $C_{k-1} = (u = c_1, c_2, ..., c_r = v)$. Let $c_p, c_q \in C_{k-1}$, respectively be the first and the last neighbor of the vertex(vertices) mapped to v_k. If only one vertex, z_0, is mapped to v_k, we update $\mathcal{U}(c_p)$, $\mathcal{U}(c_q)$ and $\mathcal{U}(z_0)$ as follows:

$$Low(c_p, c_{p+1}) = p + 1 \Longrightarrow \mathcal{U}(c_p) = \mathcal{U}(c_p) \cup \mathcal{U}(c_{p+1})$$

$$Low(c_{q-2}, c_{q-1}) = q - 2 \Longrightarrow \mathcal{U}(c_q) = \mathcal{U}(c_q) \cup \mathcal{U}(c_{q-1})$$

$$\mathcal{U}(z_0) = \mathcal{U}(z_0) \cup \bigcup_{i=Low(c_p, c_{p+1})+1}^{Low(c_{q-2}, c_{q-1})} \mathcal{U}(c_i)$$

We do not change $\mathcal{U}(g_i)$ if two vertices, z_0 and z_1, are mapped to v_k.

•*Shift to the right:* We then perform the necessary shifting. We shift each vertex $g_i \in \bigcup_{i=q}^{r} \mathcal{U}(c_i)$ to the right by one if only one vertex is mapped to v_k, by two otherwise.

•*Locate the New Vertices:* Finally we locate the vertex(vertices) mapped to v_k on the grid. Let $|v_k|$ denote the number of vertices mapped to v_k. Then we have:

If c_p has no neighbors in $G - G_k$
  $x(z_0) = x(c_p)$
  $y(z_0) = y(c_q) + x(c_q) - x(c_p) - |v_k| + 1$
otherwise
  $x(z_0) = x(c_p) + 1$
  $y(z_0) = y(c_q) + x(c_q) - x(c_p) - |v_k|$
If $|v_k| = 2$ define z_1 also:
  $x(z_1) = x(z_0) + 1$
  $y(z_1) = y(z_0)$

Up to this step, the algorithm is just a restriction of the one in [4] and it guarantees the convex drawing of the faces. Then, in order to guarantee strict-convexity, we note the following degenerate cases; see Fig. 4:

•*Degeneracies:* We check for the following:

If only one vertex, z_0, is mapped to v_k

(d_1) If $x(z_0) = x(c_{p+1}) = x(c_{p+2})$

Shift each vertex $g_i \in \bigcup_{i=p+1}^{r} \mathcal{U}(c_i)$ to the right by one.

Perform the location calculation for z_0 again.

(d_2) If $k < m$ and z_0, c_q, c_{q+1} are aligned and c_q has no neighbors in $G - G_k$

Shift each vertex $g_i \in \bigcup_{i=q+1}^{r} \mathcal{U}(c_i)$ to the right by one.

If two vertices, z_0 and z_1 are mapped to v_k

(d_3) If $y(z_0) = y(z_1) = y(c_p)$

Shift each vertex $g_i \in \bigcup_{i=q}^{r} \mathcal{U}(c_i)$ to the right by one.

Perform the location calculation for z_0 and z_1 again.

(d_4) If $k < m$ and z_1, c_q, c_{q+1} are aligned and c_q has no neighbors in $G - G_k$

Shift each vertex $g_i \in \bigcup_{i=q+1}^{r} \mathcal{U}(c_i)$ to the right by one.

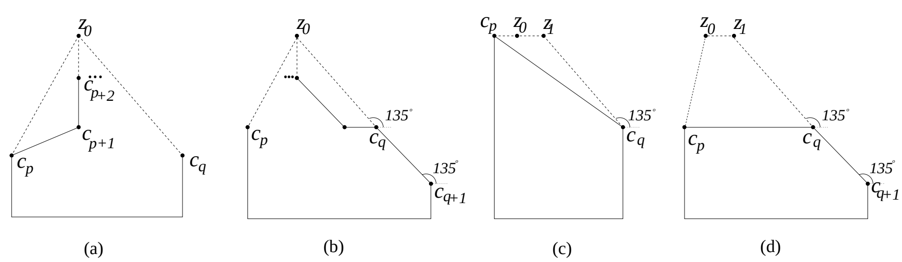

Fig. 4. Four possible degenerate cases: type d_1, type d_2, type d_3, type d_4.

2.4 Proof of Correctness

Lemma 2. *Let $C_k = (u = c_1, c_2, ..., c_r = v)$ be the exterior face of G_k after the k^{th} placement step. Let $\alpha(c_j, c_{j+1})$ denote the angle of the vector $\mathbf{c_j c_{j+1}}$, for $1 \le j \le r-1$. The following holds for $2 \le k \le m-1$:*

1. *$\alpha(c_j, c_{j+1})$ lies in $[-45°, \arctan -1/2] \cup \{0\} \cup [45°, 90°]$. It can not lie in $(-45°, \arctan -1/2]$ if c_j has a neighbor in $G - G_k$.*
2. *If $c_j \in C_k, c_j \notin \{c_1, c_r\}$ s.t. c_j does not have a neighbor in $G - G_k$, then:*
   (a) If $Low(c_{j-1}, c_j) = j-1$ then $\alpha(c_j, c_{j+1}) = 90°$; else $\alpha(c_{j-1}, c_j) = -45°$.
   (b) If $\alpha(c_j, c_{j+1}) = 90°$ then $\alpha(c_{j-1}, c_j) \neq 90°$.
   (c) If $\alpha(c_j, c_{j+1}) = -45°$ then $\alpha(c_{j-1}, c_j) \neq -45°$.

Proof Sketch: Due to space limitations the proof of this lemma is left out of this extended abstract.[1] □

Preserving Planarity Let only one vertex, z_0, be mapped to v_k. If (z_0, c_j) is an edge in G_k for some $c_j \in C_{k-1}$, then the placement algorithm and the previous lemma guarantees that $-90 < \alpha(z_0, c_j) < -45$, for $j \neq p, j \neq q$.

[1] The proof of this lemma can be found in the full version of the paper, available at `www.cs.arizona.edu/~cesim/dual.ps`.

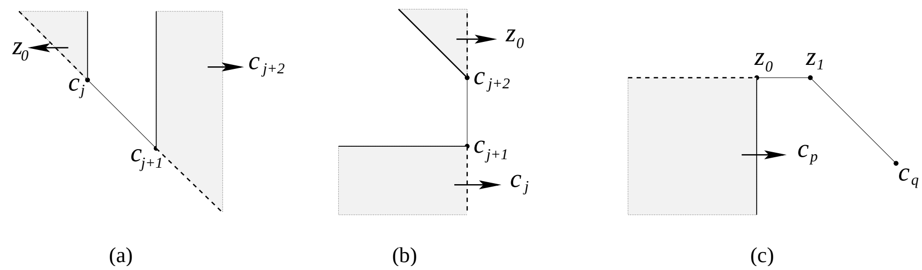

Fig. 5. The vertices pointed to by the arrows must lie in the indicated area. The dashed lines indicate open boundaries that are not included in the area.

Then no crossing is created between a new edge (z_0, c_j) and the edges of C_{k-1}. Because such a crossing would imply that there exists $j' < j$ s.t. $c_{j'} \in C_k$ and $\alpha(c_{j'}, c_j) < -45$. But this is impossible by the first part of the above lemma. The same idea applies to the case where $|v_k| = 2$. Then the following corollary holds:

Corollary 1. *Insertion of the vertex(vertices) mapped to v_k, at the k^{th} placement step, where $2 \leq k \leq m$ preserves planarity.*

Strictly Convex Faces Let $|v_k| = 1$ and z_0 be the vertex mapped to v_k. Let $\mathcal{F}_j = (c_j, c_{j+1}, c_{j+2}, z_0)$ be a quadrilateral face created after the insertion of z_0. If $Low(c_j, c_{j+1}) = j + 1$, then by the previous lemma $\alpha(c_j, c_{j+1}) = -45°$. Fig. 5(a) shows the area where z_0 and c_{j+2} must lie. If $Low(c_j, c_{j+1}) = j$, then $\alpha(c_{j+1}, c_{j+2}) = 90°$. Fig. 5(b) shows the area where z_0 and c_{j+2} must lie in this case. Both cases imply that $\mathcal{F}_j = (c_j, c_{j+1}, c_{j+2}, z_0)$ is strictly convex.

If $|v_k| = 2$ and z_0, z_1 are mapped to v_k, the placement algorithm requires that c_p must lie in the area shown in Fig. 5(c), which implies that the newly created face is strictly convex. The following corollary holds:

Corollary 2. *The newly created faces after the insertion of the vertex(vertices) mapped to v_k, at the k^{th} placement step, where $2 \leq k \leq m$, are strictly convex.*

Shifting Preserves Planarity and Strictly Convex Faces The above discussion shows that after the insertion of the vertex(vertices) at the kth placement step, no new edge crossings are created and all the newly added faces are strictly convex. In order to complete the proof of correctness we only need to prove that the same holds for shifting also:

Lemma 3. *Let $C_k = (u = c_1, c_2, ..., c_r = v)$ be the exterior face of G_k after the k^{th} placement step, where $2 \leq k < m$. For any given j, where $1 \leq j \leq r$, shifting the vertices in $\bigcup_{i=j}^{r} \mathcal{U}(c_i)$, to the right by s units preserves the planarity and the strictly convex faces of G_k.*

Proof Sketch: The claim holds trivially for $k = 2$. Assume it holds for $k' = k-1$, where $2 \leq k' < m - 1$. We assume $|v_k| = 1$. The case where $|v_k| = 2$ is similar.

Let z_0 be the vertex mapped to v_k and $c_p, c_q \in C_{k-1}$, respectively be the first and the last neighbor of $z_0$in G_{k-1}.

If $j \leq p$ then by the inductive assumption the planarity of G_{k-1} and the strictly convex faces of G_{k-1} are preserved. The faces introduced by z_0 shifts rigidly to the right, which, by the previous corollaries, implies that G_k is planar and all its faces are strictly convex.

If $j > q$, then by the inductive assumption the planarity of G_{k-1} and the strictly convex faces are preserved. Since neither z_0 nor any of its neighbors in G_{k-1} are shifted the lemma follows.

If shifting the newly inserted vertex z_0, we inductively apply the shifting to $j' = Low(c_p, c_{p+1}) + 1$ in G_{k-1}. By the inductive assumption the planarity and strictly convex faces are preserved for G_{k-1}. Since we applied a shifting starting with j' then, all the faces except the first one are shifted rigidly to the right, which implies that those faces are strictly convex. Then the only problem could arise with the leftmost face. If $Low(c_p, c_{p+1}) = p$, then c_{p+1}, c_{p+2} and z_0 are all shifted to the right by the same amount. Since initially the face $(c_p, c_{p+1}, c_{p+2}, z_0)$ was strictly convex, it continues to be so after shifting those three vertices also. In the case where $Low(c_p, c_{p+1}) = p + 1$, the only shifted vertices are z_0 and c_{p+2}. Again shifting those two vertices does not violate the convexity of the face.

If $j = q$, the situation is very similar to the previous case, except now the only deformed face is the rightmost face, instead of the leftmost one. The same idea applies to this case also, i.e., given that initially the face is strictly convex, it remains so after shifting. □

2.5 Grid Size

Lemma 4. *The algorithm requires a grid of size at most* $(2n - 4) \times (2n - 4)$.

Proof Sketch: If no degeneracies are created then the exact grid size required is $(n - 1) \times (n - 1)$. We show that each degenerate case can be associated with a newly added quadrilateral face of G.

Degenerate case of type d_1 is associated with the face $(c_p, c_{p+1}, c_{p+2}, z_0)$. Degenerate case of type d_2 at some step k of the algorithm, is associated with a face (z_0, c_q, c_{q+1}, g_i), where g_i is a vertex that will be added at some step $k' > k$ of the algorithm. We know that such a face exists, since $k < m$, c_q has no neighbors in $G - G_k$ and each face under consideration is a quadrilateral. Similar argument holds for degenerate case of type d_4. Finally degenerate case of type d_3 is associated with the face (c_p, c_q, z_1, z_0). Fig. 4 shows all four types of degeneracies that can occur. Note that each quadrilateral face is associated with at most one degeneracy.

Since an *FQ* graph G with n vertices has $n - 3$ inside faces, the placement algorithm requires grid size of at most $(2n - 4) \times (2n - 4)$. □

Final Shifting Let (u, v, w, w') be the outer face of G. The placement algorithm and Lemma-2 imply that the outer face is the isosceles right triangle $\triangle uvw'$ and that w lies on the line segment (v, w'). One final right shift is needed

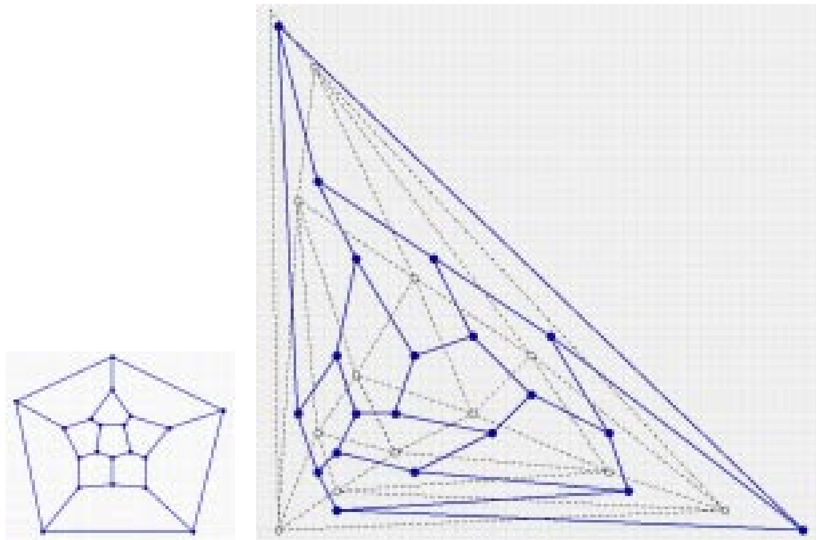

Fig. 6. Dodecahedral graph and its dual representation. The filled-in vertices and solid edges represent the primal graph; the empty circles and dashed edges represent the dual.

to guarantee that the outer face (u, v, w, w') lies on a strictly concave quadrilateral. For this we just shift v to the right by one. As a result we can draw the edge (v, w') as a straight-line segment. In order to draw the edge (u, w), we place a bend point at $(x(w')-1, y(w')+2)$, where $x(w')$ and $y(w')$, respectively denote the x and y coordinates of the vertex w'. We connect the bend point with u and w. Then the total area required is $(2n-2) \times (2n-2)$ and Theorem-1 follows.

3 Implementation

We have implemented our algorithm to visualize 3-connected planar graphs and their duals using the LEDA/AGD libraries [10]. Finding a suitable canonical labeling takes linear time [9]. We make use of the technique introduced by [5] to do the placement step. It is based on the fact that storing relative x-coordinates of the previously embedded vertices is sufficient at every step. Then the placement step also requires only linear time. Overall, the algorithm runs in linear time. Fig. 6 shows the primal/dual drawing we get for the dodecahedral graph and Fig. 7 shows the primal/dual drawing of an arbitrary 3-connected planar graph.

4 Acknowledgments

We would like to thank Anna Lubiw for introducing us to the problem of simultaneous graph embedding and for stimulating discussions about it.

References

1. M. Bern and J. R. Gilbert. Drawing the planar dual. *Information Processing Letters*, 43(1):7–13, Aug. 1992.

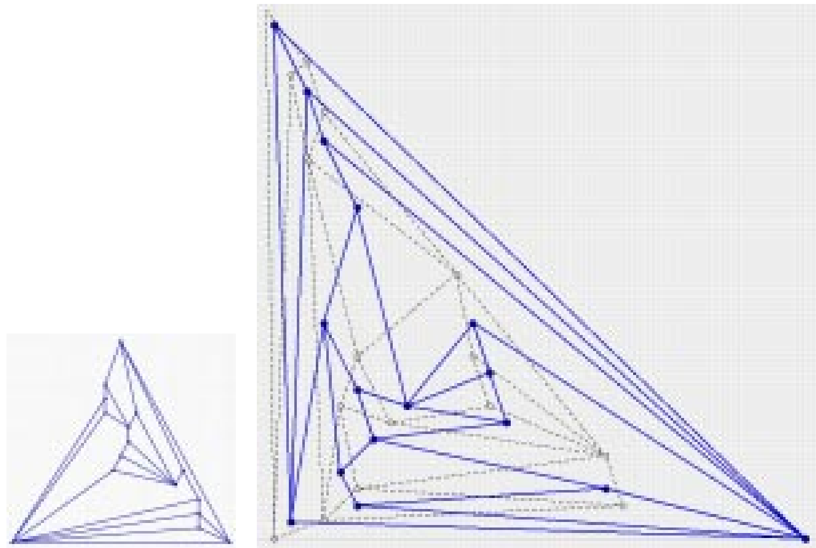

Fig. 7. A random 3-connected planar graph with 16 vertices and its dual representation.

2. G. R. Brightwell and E. R. Scheinerman. Representations of planar graphs. *SIAM Journal on Discrete Mathematics*, 6(2):214–229, May 1993.
3. M. Chrobak, M. T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 319–328, 1996.
4. M. Chrobak and G. Kant. Convex grid drawings of 3-connected planar graphs. *International Journal of Computational Geometry and Applications*, 7(3):211–223, 1997.
5. M. Chrobak and T. Payne. A linear-time algorithm for drawing planar graphs. *Inform. Process. Lett.*, 54:241–246, 1995.
6. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.
7. D. Harel and M. Sardas. An algorithm for straight-line drawing of planar graphs. *Algorithmica*, 20(2):119–135, 1998.
8. J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974.
9. G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996. (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia).
10. K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, 1999.
11. K. Miura, S.-I. Nakano, and T. Nishizeki. Grid drawings of 4-connected plane graphs. *Discrete and Computational Geometry*, 26(1):73–87, 2001.
12. B. Mohar. Circle packing of maps in polynomial time. *European Journal of Combinatorics*, 18:785–805, 1997.
13. W. Schnyder. Embedding planar graphs on the grid. In *Proceedings of the 1st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 138–148, 1990.
14. W. T. Tutte. How to draw a graph. *Proceedings London Mathematical Society*, 13(52):743–768, 1963.