Area-Efficient Drawings of Outerplanar Graphs*

Ashim Garg and Adrian Rusu

Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260
{agarg,adirusu}@cse.buffalo.edu

Abstract. We show that an outerplanar graph G with n vertices and degree d admits a planar straight-line grid drawing with area $O(dn^{1.48})$ in $O(n)$ time. This implies that if $d = o(n^{0.52})$, then G can be drawn in this fashion in $o(n^2)$ area.

1 Introduction

A drawing Γ of a graph is a *straight-line* drawing, if each edge is drawn as a single line-segment. Γ is a *grid* drawing if all the vertices have integer coordinates. Γ is a *planar* drawing, if edges do not intersect each other. Here, we concentrate on grid drawings. So, we assume that the plane is covered by an infinite rectangular grid consisting of horizontal and vertical channels. Let Γ be a grid drawing. Let R be the smallest rectangle with sides parallel to the X-and Y-axes, respectively, that covers Γ completely. The *width* (*height*) of Γ is equal to 1+ width of R (1+height of R). The *area* of Γ is equal to (1+width of R)·(1+height of R), which is equal to the number of grid points contained within R. The *degree* of a graph is equal to the maximum number of edges incident on a vertex.

There has been little work done on the area-requirement of planar straight-line grid drawings of outerplanar graphs. Currently, the best known upper bound on the area of such a drawing of an outerplanar graph with n vertices is $O(n^2)$, which is the same as for general planar graphs [3,8].

In this paper, we show that an outerplanar graph G with n vertices and degree d admits a planar straight-line grid drawing with area $O(dn^{1.48})$ in $O(n)$ time. This implies that if $d = o(n^{0.52})$, then G can be drawn in this fashion in $o(n^2)$ area.

In Section 4, we give a brief description of our drawing algorithm (for more details, see [6]). It is based on a tree-drawing algorithm of [2], and uses the fact that the dual of a maximal outerplanar graph is a tree.

2 Related Results

Let G be an outerplanar graph with n vertices. [1] shows that G admits a planar polyline drawing as well as a visibility representation with $O(n \log n)$ area. [7]

* Research supported by NSF CAREER Award IIS-9985136, NSF CISE Research Infrastructure Award No. 0101244, and Mark Diamond Research Grant No. 13-Summer-2003 from GSA of The State University of New York.

G. Liotta (Ed.): GD 2003, LNCS 2912, pp. 129–134, 2004.

shows that G admits a planar polyline drawing with $O(n)$ area, if G has degree at most 4. The technique of [7] can be easily extended to construct a planar polyline drawing of G with $O(d^2n)$ area, if G has degree d [1]. Also, in 3D, G admits a crossings-free straight-line grid drawing with $O(n)$ volume [4,5].

3 Preliminaries

We denote by $|G|$, the number of vertices (nodes) in a graph (tree) G. An *ordered* tree is one with a pre-specified counterclockwise ordering of edges incident on each node. A path $P = v_0v_1 \dots v_q$ is a *root-to-leaf* path of a binary ordered tree T, if v_0 is the root of T, and v_q is a leaf of T. A *left (right) subtree of* P is a subtree of T rooted at the left (right) child c of a node of P, such that c does not belong to P. The *size* of a subtree of T is equal to the number of nodes in it. The following lemma, which follows directly from Lemma A.1 of [2], defines the concept of a *spine*, which is a special kind of a root-to-leaf path (see also [6]).

Lemma 1 (Lemma A.1 of [2]). *Let $p = 0.48$. Given any binary ordered tree T with n nodes, there exists a root-to-leaf path P, called* spine*, such that for any left subtree α and right subtree β of P, $|\alpha|^p + |\beta|^p \le (1-\delta)n^p$, for some constant $\delta > 0$. Also, assuming that we have already pre-computed the size of the subtree rooted at each node v of T and stored it in v, we can compute P in $O(|P|)$ time.*

Let G be a *maximal* outerplanar graph, i.e., an outerplanar graph to which no edge can be added without destroying its outerplanarity. It is easy to see that each internal face of G is a triangle. The *dual tree* T_G of G is defined as follows:

– there is a one-to-one correspondence between the nodes of T_G and the internal faces of G, and
– there is an edge $e = (u, v)$ in T_G if and only if the faces of G corresponding to u and v share an edge e' on their boundaries. e and e' are *duals* of each other.

(Figure 1(b) shows the dual tree of the outerplanar graph of Figure 1(a).)

Let $P = v_0v_1 \dots v_q$ be a path of T_G. Let H be the subgraph of G corresponding to P. A *beam* drawing of H is shown in Figure 2, where the vertices of H are placed on two horizontal channels, and the faces of H are drawn as triangles.

A line-segment with end-points a and b is a *flat* line-segment if a and b are grid points, and either belong to the same horizontal channel, or belong to adjacent horizontal channels. Let B be a flat line-segment with end-points a and b, such that b is at least one unit to the right of a. Let G be an outerplanar graph with two distinguished adjacent vertices u and v, such that the edge (u, v) is on the external face of G; u and v are called the *poles* of G. Let D be a planar straight-line drawing of G. D is a *feasible* drawing of G with base B if:

– the two poles of G are mapped to a and b each,
– each non-pole vertex of G is placed at least one unit above the lower of a and b, at least one unit to the right of a, and at least one unit to left of b.

Throughout the rest of this paper, for simplicity, by the term *outerplanar* graph, we will mean a maximal outerplanar graph.

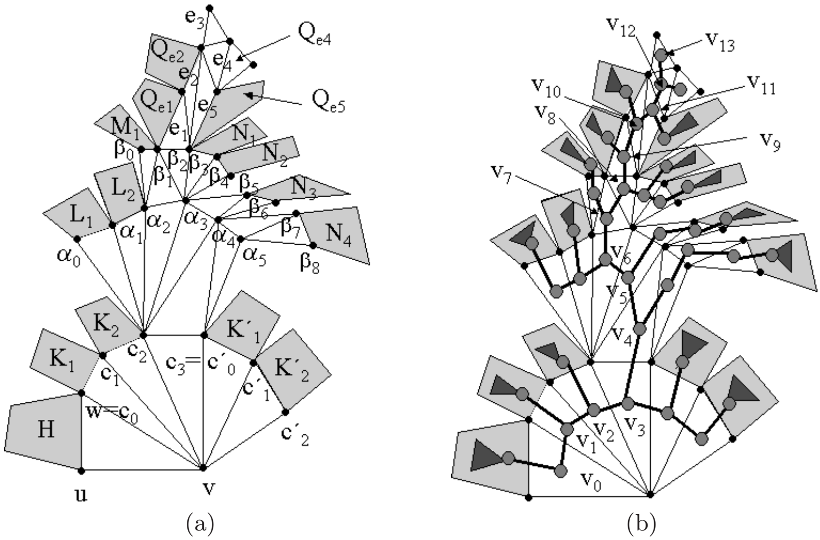

Fig. 1. (a) An outerplanar graph G. Here, H, K_1, K_2, K_1', K_2', L_1, L_2, M_1, N_1, N_2, N_3, N_4, Q_{e1}, Q_{e2}, Q_{e4}, and Q_{e5} are subgraphs of G, and are themselves outerplanar graphs. (b) The dual tree T_G of G. The edges of T_G are shown with dark lines. Note that $v_0 v_1 \ldots v_{13}$ is a spine of T_G.

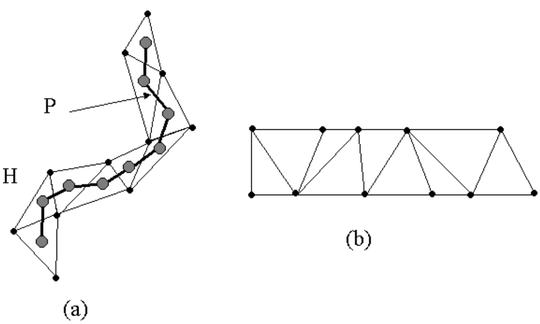

Fig. 2. (a) A path P and its corresponding graph H. (b) A beam drawing of H.

4 Outerplanar Graph Drawing Algorithm

The drawing algorithm, which we call *Algorithm OpDraw*, is recursive in nature. In each recursive step, it takes as inputs an outerplanar graph G with prespecified poles, and a long-enough flat line-segment B, and constructs a feasible drawing D of G with base B. D is constructed by constructing a drawing M of the subgraph Z corresponding to a spine of T_G, splitting G into several smaller outerplanar graphs after removing Z and some other vertices from it, constructing feasible drawings of these smaller outerplanar graphs, and then combining their drawings with M to obtain D.

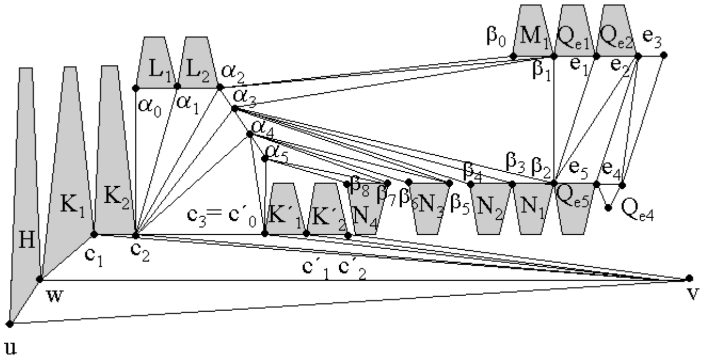

Fig. 3. The drawing of the outerplanar graph of Figure 1(a) constructed by *Algorithm OpDraw*, in the case, where v is one unit above u.

Let u and v be the poles of G. Let T_G be the dual tree of G. Let r be the node of T_G that corresponds to the internal face F of G that contains the edge (u, v). Convert T_G into an ordered tree by making r its root, and assigning the edges incident on each node o the same counterclockwise order as the one that their dual edges have in the face of G corresponding to o. Note that T_G is a binary tree because each internal face of G is a triangle.

Let $P = v_0v_1v_2 \ldots v_q$ be a spine of T_G, where $v_0 = r$. In general, we can interpret the structure of G with respect to P as follows (see Figure 1): Let u, v, w be the vertices belonging to face F. Assume that the edge (v_0, v_1) of P is the dual of the edge (v, w) (the case, where (v_0, v_1) is the dual of (u, w), is symmetrical). Let $u, w = c_0, c_1, \ldots, c_m = c'_0, c'_1, \ldots, c'_s$ be the clockwise order of the neighbors of v, where m is the integer such that for each i $(1 \le i \le m)$, the face $c_{i-1}c_iv$ corresponds to the spine node v_i, and for each i $(1 \le i \le s)$, the face $c'_{i-1}c'_iv$ corresponds to a non-spine node of T_G (In Figure 1, $m = 3$, and $s = 2$). Let H, K_i, and K'_i be the maximal biconnected subgraphs of G that contains edges (u, w), (c_{i-1}, c_i), and (c'_{i-1}, c'_i), respectively, but not the faces uvw, $vc_{i-1}c_i$, and $vc'_{i-1}c'_i$, respectively. Let $\alpha_0, \alpha_1, \ldots, \alpha_h, \alpha_{h+1}, \alpha_t$ be the vertices of K_m different from c_{m-1} and c_m such that $\alpha_0, \alpha_1, \ldots, \alpha_h$ is the clockwise order of the neighbors of c_{m-1}, and $\alpha_h, \alpha_{h+1}, \alpha_{h+2}, \alpha_t$ is the clockwise order of the neighbors of c_m. For example, in Figure 1, $h = 4$, and $t = 5$. Let j be the index such that the dual of edge (α_{j-1}, α_j) belongs to P (if no such j exists, then we can do the following: if K_m consists of only one internal face, namely, $c_{m-1}c_m\alpha_0$, then set $j = 0$. Otherwise, the leaf v_q of P will correspond to either the face $\alpha_0\alpha_1c_{m-1}$ or the face $\alpha_{t-1}\alpha_tc_m$; in the first case, set $j = 1$, and in the second case, set $j = t$). For example, in Figure 1, $j = 3$. Let L_i be the maximal biconnected subgraph of G that contains the edge (α_{i-1}, α_i), but not the face $\alpha_{i-1}\alpha_ic_{m-1}$ or $\alpha_{i-1}\alpha_ic_m$ (whichever exists). Let $S = \beta_0, \beta_1, \ldots, \beta_\mu$ be the clockwise order of the neighbors of $\alpha_{j-1}, \alpha_j, \ldots, \alpha_t$ (in that order) in the subgraphs $L_j, L_{j+1}, \ldots, L_t$, where each β_k is different from $\alpha_{j-1}, \alpha_j, \ldots, \alpha_t$. For example, in Figure 1, $\mu = 8$. Let ϵ be the index such that the dual of the edge $(\beta_{\epsilon-1}, \beta_\epsilon)$ belongs to P (if no such

ϵ exists, then we can do the following: if L_j consists of only one internal face, namely, $\alpha_{j-1}\alpha_j\beta_0$, then set $\epsilon = 0$. Otherwise, the leaf v_q of P will correspond to either the face $\beta_0\beta_1\alpha_{j-1}$ or the face $\beta_{\mu-1}\beta_\mu\alpha_j$; in the first case, set $\epsilon = 1$, and in the second case, set $\epsilon = \mu$). For example, in Figure 1, $\epsilon = 2$. For each i, where $1 \leq i \leq \epsilon - 1$ ($\epsilon + 1 \leq i \leq \mu$), if there is an edge (β_{i-1}, β_i), then let M_i ($N_{i-\epsilon}$) be the maximal biconnected subgraph of G that contains (β_{i-1}, β_i), but not the face $\beta_{i-1}\beta_i\alpha_k$, where $k = j - 1$ or j. Let $(v_{\rho-1}, v_\rho)$ be the edge of P that is the dual of the edge $(\beta_{\epsilon-1}, \beta_\epsilon)$. For example, in Figure 1, $\rho = 9$. Let R be the subgraph of G that corresponds to the subpath $v_\rho v_{\rho+1} \ldots v_q$ of P. Let $e \neq (\beta_{\epsilon-1}, \beta_\epsilon)$ be an edge on the external face of R. Let Q_e be the maximal biconnected subgraph of G than contains e but not the face of R containing e.

D is constructed as shown in Figure 3 (in this figure, we show the construction when v is one unit above u. The other two cases, where u is one unit above v, and where u and v are in the same horizontal channel are similar. For more details, see [6]): w is placed one unit above u, and $c_1, c_2, \ldots, c_m = c'_0, c'_1, c'_s$ are placed in the same horizontal channel one unit above w. α_0 (α_t) is placed in the same vertical channel as c_{m-1} (c_m). $\alpha_0, \alpha_1, \ldots, \alpha_{j-1}$ are placed in the same horizontal channel. $\alpha_{j-1}, \alpha_j, \ldots, \alpha_t$ are placed along a line making 45° with the horizontal channels, such that each α_k is one unit above and one unit to the left of α_{k+1}. $\beta_0, \beta_1, \ldots, \beta_{\epsilon-1}$ are placed one unit above α_{j-1} in the same horizontal channel. $\beta_\epsilon, \beta_{\epsilon+1}, \ldots, \beta_\mu$ are placed one unit below α_t in the same horizontal channel. $\beta_{\epsilon-1}$ and β_ϵ are placed in the same vertical channel. A beam drawing E of R is constructed. Feasible drawings of H with base $\overline{uw}$, and each K_i ($1 \leq i \leq m-1$), K'_i ($1 \leq i \leq s$), L_i ($1 \leq i \leq j-1$), M_i ($1 \leq i \leq \epsilon - 1$), N_i ($1 \leq i \leq \mu - \epsilon$), and Q_e, with bases $\overline{c_{i-1}c_i}$, $\overline{c'_{i-1}c'_i}$, $\overline{\alpha_{i-1}\alpha_i}$, $\overline{\beta_{i-1}\beta_i}$, $\overline{\beta_{i+\epsilon-1}\beta_{i+\epsilon}}$, and e, respectively, are recursively constructed, with the horizontal distances between the end-points of the line-segments $\overline{uw}$, $\overline{c_{i-1}c_i}$, $\overline{c'_{i-1}c'_i}$, $\overline{\alpha_{i-1}\alpha_i}$, $\overline{\beta_{i-1}\beta_i}$, $\overline{\beta_{i+\epsilon-1}\beta_{i+\epsilon}}$, and e, equal to $|H|-1$, $|K_i|-1$, $|K'_i|-1$, $|L_i|-1$, $|M_i|-1$, $|N_i|-1$, and $|Q_e|-1$, respectively. The drawing of each N_i, and Q_e, where e is on the bottom boundary of E, is flipped upside-down before placing it in D. Also note that β_0 and β_μ are placed such that they are either in the same vertical channel as, or to the right of c'_s. Also note that α_t is placed $1 + \theta$ units above the horizontal channel containing c'_s, where θ is maximum height of the feasible drawings of K'_i, N_i, and Q_e, where e is on the bottom boundary of E.

Let $h(n)$ and $w(n)$ be the height and width, respectively, of D, as constructed by Algorithm *OpDraw*. Here, n is the number of vertices in G. Let d be the degree of G. Note that, by the definition of feasible drawings, $w(n)$ will be equal to one plus the horizontal separation between the end-points of B.

It is easy to prove using induction that $w(n) = n$ is sufficient. As for $h(n)$, first notice that, because G has degree d, $t - (j-1)$ is less than $2d$, and hence, the distance between $\beta_{\epsilon-1}$ and β_ϵ is less than $2d + 2$. Let h' be a function, such that $h'(f) = h(n)$, where f is the number of internal faces in G, i.e., the number of nodes in the dual tree T_G of G. From the construction of D, we have that:

$$h'(f) \leq \max\{\max_{1 \leq i \leq \mu-\epsilon}\{h'(|T_{N_i}|)\}, \max_{edge\ e\ on\ bottom\ boundary\ of\ E}\{h'(|T_{Q_e}|)\},$$

$$\max_{1\le i\le s}\{h'(|T_{K'_i}|)\}\} + \max\{h'(|T_H|), \max_{1\le i\le m-1}\{h'(|T_{K_i}|)\}, \max_{1\le i\le j-1}\{h'(|T_{L_i}|)\},$$
$$\max_{1\le i\le \epsilon-1}\{h'(|T_{M_i}|)\}, \max_{edge\ e\ on\ top\ boundary\ of\ E}\{h'(|T_{Q_e}|)\}\} + O(d),$$

The dual trees of H, K_i, L_i, M_i, and Q_e (where edge e is on top boundary of E) are either right subtrees of P, or belong to right subtrees of P. The dual trees of K'_i, N_i, and Q_e (where edge e is on bottom boundary of E) are either left subtrees, or belong to left subtrees of P. Hence, from Lemma 1, it follows that:

$$h'(f) \le \max_{f_1^p + f_2^p \le (1-\delta)f^p}\{h'(f_1) + h'(f_2) + O(d)\}.$$

Using induction, we can show that $h'(f) = O(df^{0.48})$ (see also [2]). Since $f = O(n)$, $h(n) = h'(f) = O(df^{0.48}) = O(dn^{0.48})$.

Theorem 1. *Let G be an outerplanar graph with degree d and n vertices. G admits a planar straight-line grid drawing with area $O(dn^{1.48})$ in $O(n)$ time.*

Proof. Arbitrarily select any edge $e = (u, v)$ on the external face of G, and designate u and v as the poles of G. Let B be any horizontal line-segment with length $n - 1$, such that the end-points of B are grid points. Construct a feasible drawing D of G with base B using Algorithm *OpDraw*. From the discussion given above, it follows immediately that the area of D will be equal to $n \cdot O(dn^{0.48}) = O(dn^{1.48})$. Using Lemma 1, we can easily implement Algorithm *OpDraw* such that it will run in $O(n)$ time.

Corollary 1. *Let G be an outerplanar graph with n vertices and degree d, where $d = o(n^{0.52})$. G admits a planar straight-line grid drawing with $o(n^2)$ area in $O(n)$ time.*

References

1. T. Biedl. Drawing outer-planar graphs in $O(n \log n)$ area. In *Proc. 10th Int. Symp. on Graph Drawing*, volume 2528 of *LNCS*, pages 54–65. Springer-Verlag, 2002.
2. T.M. Chan. A near-linear area bound for drawing binary trees. *Algorithmica*, 34(1):1–13, 2002.
3. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.
4. V. Dujmovic and D.R. Wood. Tree-partitions of k-trees with applications in graph layout. In *Proc. 29th Workshop on Graph Theoretic Concepts in Computer Science (WG '03).* To appear.
5. S. Felsner, G. Liotta, and S. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. In *Proc. 9th Int. Symp. on Graph Drawing (GD '01)*, volume 2265 of *LNCS*, pages 328–342. Springer-Verlag, 2001.
6. A. Garg and A. Rusu. Area-efficient drawings of outerplanar graphs. Technical Report No. 2003-11, Dept. Computer Sc. & Engg., Univ. at Buffalo, Buffalo, 2003.
7. C. E. Leiserson. Area-efficient graph layouts (for VLSI). In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 270–281, 1980.
8. W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 138–148, 1990.