Plane 3-trees: Embeddability & Approximation

(Extended Abstract)

Stephane Durocher* and Debajyoti Mondal†

Department of Computer Science, University of Manitoba
{durocher,jyoti}@cs.umanitoba.ca

Abstract. We give an $O(n \log^3 n)$-time linear-space algorithm that, given a plane 3-tree G with n vertices and a set S of n points in the plane, determines whether G has a point-set embedding on S (i.e., a planar straight-line drawing of G where each vertex is mapped to a distinct point of S), improving the $O(n^{4/3+\varepsilon})$-time $O(n^{4/3})$-space algorithm of Moosa and Rahman. Given an arbitrary plane graph G and a point set S, Di Giacomo and Liotta gave an algorithm to compute 2-bend point-set embeddings of G on S using $O(W^3)$ area, where W is the length of the longest edge of the bounding box of S. Their algorithm uses $O(W^3)$ area even when the input graphs are restricted to plane 3-trees. We introduce new techniques for computing 2-bend point-set embeddings of plane 3-trees that takes only $O(W^2)$ area. We also give approximation algorithms for point-set embeddings of plane 3-trees. Our results on 2-bend point-set embeddings and approximate point-set embeddings hold for partial plane 3-trees (e.g., series-parallel graphs and Halin graphs).

1 Introduction

A *planar drawing* of a graph G is an embedding (i.e., a mapping) of G onto the Euclidean plane $\mathbb{R}^2$, where each vertex in G is assigned a unique point in $\mathbb{R}^2$ and each edge in G is a simple curve in $\mathbb{R}^2$ joining the points corresponding to its endvertices such that no two curves intersect except possibly at their endpoints. A graph is *planar* if it has a planar drawing. A *straight-line drawing* of a planar graph is a planar drawing, where each edge is drawn as a straight line segment. The straight-line drawing style is popular since it naturally produces drawings that are easier to read and to display on smaller screens [1, 2]. To meet the requirements of different practical applications, researchers have examined the straight-line drawing problem under various constraints, e.g., when the vertices are constrained to be placed on a set of pre-specified locations [3, 4]. If the pre-specified locations for placing the vertices of the input graph are points on the Euclidean plane, then we call the problem a point-set embedding problem. Such problems have applications in VLSI circuit layout, where different circuits need to be mapped onto a fixed printed circuit board, simultaneous display of different

* Work of the author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

† Work of the author is supported in part by a University of Manitoba Graduate Fellowship.

social and biological networks, and construction of a desired network among a set of fixed locations. Formally, a *point-set embedding* of a *plane graph* G (i.e., a fixed combinatorial planar embedding of G) with n vertices on a set S of n points is a straight-line drawing of G, where the vertices are placed on distinct points of S.

Point-Set Embeddings. In 1994, Ikebe et al. [5] gave an $O(n^2)$-time algorithm to embed any tree with n vertices on any set of n points in *general position*, i.e., no three points are collinear. Later, Bose et al. [6] devised a divide and conquer algorithm that runs in $O(n \log n)$ time. In 1996, Castañeda and Urrutia [7] gave an $O(n^2)$-time algorithm to construct point-set embeddings of maximal outerplanar graphs. Later, Bose [3] improved the running time of their algorithm to $O(n \log^3 n)$ using a dynamic convex hull data structure. In the same paper Bose posed an open problem that asks to determine the time complexity of testing the point-set embeddability for planar graphs. In 2006, Cabello [4] proved the problem to be NP-complete for graphs that are 2-connected and 2-outerplanar. The problem remains NP-complete for 3-connected planar graphs [8], even when the treewidth is constant [9].

In the last few years researchers have examined the point-set embeddability problem restricted to *plane* 3*-trees* (also known as stacked polytopes, Apollonian networks, and maximal planar graphs with treewidth three) because of their wide range of applications in many theoretical and applied fields [10]. Nishat et al. [11] first gave an $O(n^2)$-time algorithm for deciding point-set embeddability of plane 3-trees, and proved an $\Omega(n \log n)$-time lower bound. Later, Durocher et al. [12] and Moosa and Rahman [13] independently improved the running time to $O(n^{4/3+\varepsilon})$, for any $\varepsilon > 0$. Since $\Omega(n^{4/3})$ is a lower bound on the worst-case time complexity for solving various geometric problems [14], it may be natural to accept the possibility that the $O(n^{4/3+\varepsilon})$-time algorithm could be asymptotically optimal. In fact, Moosa and Rahman mention that an $o(n^{4/3})$-time algorithm seems unlikely using currently known techniques. However, in this paper we prove that the $\Omega(n \log n)$ lower bound is nearly tight, giving an $O(n \log^3 n)$-time algorithm for deciding point-set embeddability of plane 3-trees.

Universal Point Set. Observe that a planar graph may not always admit point-set embedding on a given point set. Attempts have been made at constructing a set S of $k \geq n$ points such that every planar graph with n vertices admits a point-set embedding on a subset of S [15, 16]. Such a point set that *supports* all planar graphs with n vertices is called a *universal point set for* n. A long standing open question in graph drawing asks to design a set of $O(n)$ points that is universal for all planar graphs with n vertices [15]. Recently, Everett et al. [16] have designed a 1*-bend universal point set* S_n for planar graphs with n vertices, i.e., every planar graph with n vertices admits a straight-line drawing on S_n such that each vertex is mapped to a distinct point and each edge is drawn as a chain of at most two straight line segments.

The point-set embeddability problem seems to have close relation with the universal point set problem. Castañeda and Urrutia [7] proved that any set of n points in general position is universal for all outerplanar graphs with n vertices.

Later, Kaufmann and Wiese [17] proved that any set S of n points is 2-*bend universal* for n (i.e., every planar graph with n vertices admits a straight-line drawing on S such that each vertex is mapped to a distinct point and each edge is drawn as a chain of at most three straight line segments). However, the area required for the drawing could be exponential in W, where W is the length of the side of the smallest axis-parallel square that encloses S. Di Giacomo and Liotta [18, Theorem 7] showed that using the concept of monotone topological book embeddingone can reduce the area requirement to $O(W^3)$. Even when restricted to simpler classes of graphs (e.g., series parallel graphs or plane 3-trees), the technique of Di Giacomo and Liotta is the best known, which still requires $O(W^3)$ area. In this paper, we contribute a new technique that uses only $O(W^2)$ area to compute 2-bend point set embeddings of plane 3-trees, and hence also for partial plane 3-trees (e.g., series-parallel graphs and Halin graphs).

Approximate Point-Set Embeddings. Although any set of n points in general position is universal for n-vertex outerplanar graphs [7], a plane 3-tree with n vertices may not admit a point-set embedding on a given set of n points [11]. On the other hand, while allowing two bends per edge, any set of n points in general position is 2-bend universal for plane 3-trees. Due to this apparent difficulty of defining algorithms that simultaneously minimize area, the number of bends, and running time, we consider algorithms that provide approximate solutions, that is, at least a fraction ρ of the vertices of the input graph are mapped to distinct points of the given point set. Specifically, if the input points are in general position, then we prove that the point-set embeddability of plane 3-trees is approximable with factor $\Omega(1/\sqrt{n})$.

2 Faster Point-Set Embeddings of Plane 3-Trees

In this section we give an $O(n \log^3 n)$-time algorithm for deciding point-set embeddability of plane 3-trees. Before going into details, we review a few definitions.

A *plane* 3-*tree* G with $n \geq 3$ vertices is a triangulated plane graph such that if $n > 3$, then G contains a vertex whose deletion yields a plane 3-tree with $n-1$ vertices. Let r, s, t be a cycle of three vertices in G. By G_{rst} we denote the subgraph induced by r, s, t and the vertices that lie interior to the cycle. Every plane 3-tree G with $n > 3$ vertices contains a vertex that is the common neighbor of all the three outer vertices of G. We call this vertex the *representative vertex* of G. Let p be the representative vertex of G and let a, b, c be the three outer vertices of G in clockwise order. Then each of the subgraphs G_{abp}, G_{bcp} and G_{cap} is a plane 3-tree. Let S be a set of n points in the plane. Let p, q and r be three points that do not necessarily belong to S. Then $S(pqr)$ consists of the points of S that lie either on the boundary or in the interior of the triangle pqr.

Overview of Known Algorithms. Let G be a plane 3-tree with n vertices, and let a, b, c and p be the three outer vertices and the representative vertex of G, respectively. Nishat et al. [11]'s algorithm is as follows.

Step 1. If the number of points on the boundary of the convex hull C of S is not exactly three, then G does not admit a point-set embedding on S. Otherwise, let x, y, z be the points on C.
Step 2. For each of the possible six different mappings of the outer vertices a, b, c to the points x, y, z, execute Step 3.
Step 3. Let n_1, n_2 and n_3 be the number of vertices of G_{abp}, G_{bcp} and G_{cap}, respectively. Without loss of generality assume that the current mapping of a, b and c is to x, y and z, respectively. Find the unique mapping of the representative vertex p of G to a point $w \in S$ such that the triangles xyw, yzw and zxw properly contain exactly n_1, n_2 and n_3 points, respectively. If no such mapping of p exists, then G does not admit a point-set embedding on S for the current mapping of a, b, c; hence go to Step 2 for the next mapping. Otherwise, recursively compute point-set embeddings of G_{abp}, G_{bcp} and G_{cap} on $S(xyw), S(yzw)$ and $S(zxw)$, respectively. See Figures 1(a)–(d).

Observe that the recurrence relation for the time taken in Step 3 is $T(n) = T(n_1) + T(n_2) + T(n_3) + \mathcal{T}$, where $\mathcal{T}$ denotes the time required to find the mapping of the representative vertex. The algorithm of Nishat et al. [11] preprocesses the set S in $O(n^2)$ time so that the computation for the mapping of a representative vertex takes $O(n)$ time. Hence $\mathcal{T} = O(n)$ and the overall time complexity becomes $O(n^2)$. Moosa and Rahman [13] used a binary search technique with the help of a triangular range search data structure of Chazelle et al. [19] to obtain $\mathcal{T} = \min\{n_1, n_2, n_3\} \cdot n^{1/3+\varepsilon}$ and $T(n) = O(n^{4/3+\varepsilon})$. Durocher et al. [12] use the same idea, but instead of a binary search they use a randomized search.

Embedding Plane 3-Trees in $O(n \log^3 n)$ time. We speed up the mapping of the representative vertex as follows. We first select $O(\min\{n_1, n_2, n_3\})$ points interior to the triangle xyz in $O(\min\{n_1 + n_2, n_2 + n_3, n_1 + n_3\} \log^2 n)$ time using a dynamic convex hull data structure. We prove that these are the only candidates for the mapping of the representative vertex. We then make some non-trivial observations to test and compute a mapping for the representative vertex in $O(\min\{n_1, n_2, n_3\})$ time. Hence we obtain $\mathcal{T} = O(\min\{n_1 + n_2, n_2 + n_3, n_1 + n_3\} \log^2 n)$ and a running time of $T(n) = O(n \log^3 n)$, which dominates the $O(n \log^2 n)$ time for building the initial dynamic convex hull data structure.

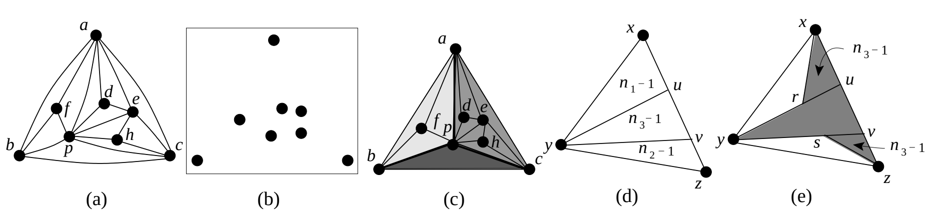

Fig. 1. (a) A plane 3-tree G. (b) A point set S. (c) A valid mapping of the representative vertex of G, and the recursive computation of the three subproblems. (d)–(e) Illustration for the lines uy, vy, xr and zs. The region of interest is shown in gray.

In the following we use three lemmas to obtain our main result. Lemma 1 selects a region R containing the candidate points inside the triangle xyz. Lemma 2 reduces the problem of finding a mapping inside the triangle xyz to the problem of finding a point satisfying specific criteria inside R. Lemma 3 gives an efficient technique to find such a point. Finally, we use these lemmas to obtain a mapping for the representative vertex in $O(\min\{n_1+n_2, n_2+n_3, n_1+n_3\}\log^2 n)$ time.

Without loss of generality assume that $n_3 \le n_2 \le n_1$. Observe that $n_1 + n_2 + n_3 - 5 = n$. Let S be a set of n points in general position such that the convex hull of S contains exactly three points x, y, z on its boundary. Without loss of generality assume that the vertices outer vertices a, b, c are mapped to the points x, y, z, respectively.

Let u and v be two points on the straight line segment xz such that $|S(uxy)| = n_1 - 1$ and $|S(vzy)| = n_2 - 1$, as shown in Figure 1(d). It is straightforward to verify that if a *valid mapping* for the representative vertex exists (i.e, there exists a point $w \in S$ such that $|S(wxy)| = n_1$, $|S(wyz)| = n_2$ and $|S(wzx)| = n_3$), then the corresponding point (i.e., the point w) must lie inside $S(uvy)$. Let r and s be two points on the straight line segments uy and vy, respectively, such that $|S(rux)| = |S(svz)| = n_3 - 1$. We call the region defined by the simple polygon x, u, v, z, s, y, r, x the *region of interest*. An example is shown in Figures 1(e). We will use the following lemma whose proof is omitted due to space constraints.

Lemma 1. *If there exists a point $w \in S$ that corresponds to a valid mapping for the representative vertex of G, then the straight line segments wx, wy and wz lie inside the region of interest R. Moreover, the number of points in R that belong to S is $O(n_3)$, and the following properties hold.*
(a) *If the points s, y, z (respectively, points r, x, y) are distinct, then $|S(syz)| = n_2 - n_3 + 2$ (respectively, $|S(rxy)| = n_1 - n_3 + 2$).*
(b) *Otherwise, point s (respectively, point r) coincides with y (respectively, y) and $|S(syz)| = 2$ (respectively, $|S(rxy)| = 2$).*

Let $S' \subseteq S$ be the set that consists of the points lying on the boundary of R and the points lying in the proper interior of R. We call S' the *set of interest*. By Lemma 1, $|S'|{=}O(n_3)$. We reduce the problem of finding a valid mapping in S to the problem of finding a point with certain properties in S', as shown in the following lemma. We omit its proof due to space constraints.

Lemma 2. *There exists a valid mapping for the representative vertex of G in S if and only if there exists a point $w' \in S'$ such that $|S'(w'yz)| = n_2 - |S(yzs)| + 3$, $|S'(w'xy)| = n_1 - |S(xyr)| + 3$ and $|S'(w'xz)| = n_3$.*

Since a valid mapping for the representative vertex is unique, w' must be unique. We call the point w' the *principal point* of S'. Observe that this principal point corresponds to the valid mapping of the representative vertex of G in S.

Lemma 3. *Let S be a set of $t \ge 4$ points in general position such that the convex hull of S is a triangle xyz. Let i, j, k be three non-negative integers, where $i \ge 3, j \ge 3$ and $k = t + 5 - i - j$. Then we can decide in $O(t)$ time whether there exists a point $w \in S$ such that $|S(wxy)| = i, |S(wyz)| = j$ and $|S(wxz)| = k$, and compute such a point if it exists.*

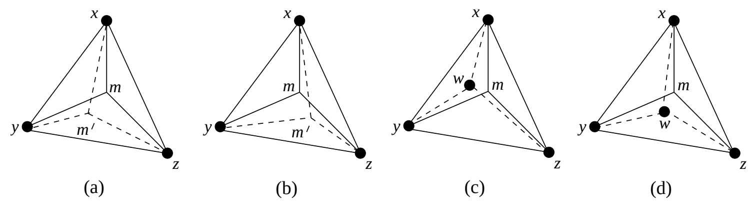

Fig. 2. Illustration for the proof of Lemma 3, where $\{m, m'\} \cap S = \emptyset$ and $\{x, y, z, w\} \subset S$.

Proof. Consider first a variation of the problem, where we want to construct a point $m \notin S$ interior to xyz such that $|S(mxy)| = i + 1, |S(myz)| = j - 1$ and $|S(mxz)| = k - 1$. Steiger and Streinu [20] proved the existence of m and gave an $O(t)$-time algorithm to find m. If there exists a point $w \in S$ such that $|S(wxy)| = i, |S(wyz)| = j$ and $|S(wxz)| = k$, then it is straightforward to observe that there exists a point $m \notin S$ interior to xyz such that $|S(mxy)| = i+1, |S(myz)| = j-1$ and $|S(mxz)| = k - 1$. We now prove that the existence of m implies a unique partition of S. Hence we can efficiently test whether w exists.

We claim that if there exists a point $m' \neq m$, where $m' \notin S$, such that $|S(m'xy)| = i + 1, |S(m'yz)| = j - 1$ and $|S(m'xz)| = k - 1$, then the sets $S(m'xy), S(m'yz)$ and $S(m'xz)$ must coincide with the sets $S(mxy), S(myz)$ and $S(mxz)$. To verify the claim assume without loss of generality that $m' \in S(myz)$. Since the triangle $m'yz$ lies interior to the triangle myz, the sets $S(m'yz)$ and $S(myz)$ must be identical. On the other hand, either the triangle mxz lies interior to the triangle $m'xz$, or the triangle mxy lies interior to the triangle $m'xy$, as shown in Figures 2(a)–(b). Therefore, either the sets $S(mxz)$ and $S(m'xz)$, or the sets $S(mxy)$ and $S(m'xy)$ must be identical. Consequently, the remaining pair of sets must also be identical.

Observe that if the point $w \in S$ we are looking for exists, then w must lie interior to $S(mxy)$, as shown in Figure 2(c). Otherwise, if $w \in S(myz)$ (respectively, $w \in S(mxz)$), then $|S(myz)| \geq |S(wyz)| = j$ (respectively, $|S(mxz)| \geq |S(wxz)| = k$), which contradicts our initial assumption that $|S(myz)| = j - 1$ (respectively, $|S(mxz)| = k - 1$). Figure 2(d) depicts such a scenario. If w exists, then the convex hull of $S(mxy)$ must be a triangle xym'', where $m'' \in S(mxy)$. If $|S(m''xy)| = i, |S(m''yz)| = j$ and $|S(m''xz)| = k$, then m'' is the required point w. Otherwise, no such w exists.

We can test whether the convex hull of $S(mxy)$ is a triangle in $O(t)$ time (e.g., find the leftmost point a, the rightmost point b and the point c with the largest perpendicular distance to the line determined by the line segment ab, and then test whether triangle abc contains all the points). It is also straightforward to compute the values $|S(m''xy)|, |S(m''yz)|$ and $|S(m''xz)|$ in $O(t)$ time. □

Given the set of interest $S' \subseteq S$, we use Lemmas 2 and 3 to find the principal point $w' \in S'$ in $O(n_3)$ time. Observe that this principal point corresponds to the valid mapping of the representative vertex of G in S. We now show how to compute the set S' in $O((n_2 + n_3) \log^2 n)$ time using the dynamic planar convex hull data structure of Overmars and van Leeuwen [21], which supports a single update (i.e., a single insertion or deletion) in $O(\log^2 n)$ time.

Step A. Assume that the points of S are placed in a dynamic convex hull data structure $\mathcal{D}$. We recursively delete the neighbor of y on the boundary of the convex hull of S starting from z in anticlockwise order. After deleting $n_2 - 2$ points, we insert all the deleted points into a new dynamic convex hull data structure $\mathcal{D}'$. We then insert a copy of y into $\mathcal{D}'$. Recall u and v from Figure 1(e). Observe that all the points of $S(vyz)$ are placed in $\mathcal{D}'$. In a similar way we construct another dynamic convex hull data structure $\mathcal{D}''$ that maintains all the points of $S(uvy)$. Consequently, $\mathcal{D}$ now only maintains the points of $S(uxy)$. Since a single insertion or deletion takes $O(\log^2 n)$ time, all the above $O(n_2+n_3)$ insertions and deletions take $O((n_2+n_3)\log^2 n)$ time in total.

Step B. We now construct two other dynamic convex hull data structures $\mathcal{D}_1$ and $\mathcal{D}_2$ using $\mathcal{D}$ and $\mathcal{D}'$ such that they maintain the points of $S(rux)$ and $S(svz)$, respectively. Since $|S(rux)| + |S(svz)| = O(n_3)$, this takes $O(n_3 \log^2 n)$ time.

Step C. We construct the point set S' using the points maintained in $\mathcal{D}'', \mathcal{D}_1$ and $\mathcal{D}_2$, which also takes $O(n_3 \log^2 n)$ time. In similar way we can restore the original point set S and the initial data structure $\mathcal{D}$ in $O((n_2+n_3)\log^2 n)$ time.

The time for the construction of S' using Steps A–C is $O((n_2+n_3)\log^2 n)$, which dominates the time required for the computation of the valid mapping of the representative vertex p. Let w be the point that corresponds to the valid mapping. We now need to construct the point sets $S(wxy), S(wyz)$ and $S(wzx)$ for recursively testing the point-set embeddability of G_{abp}, G_{bcp} and G_{cap}, respectively. We can construct $S(wxy), S(wyz)$ and $S(wzx)$ and their corresponding dynamic convex hull data structures in $O((n_2+n_3)\log^2 n)$ time as follows. Let l be the point of intersection of the straight lines determined by the line segments wy and xz. First construct the set $S(lyz)$ and then modify it to obtain the sets $S(wyz)$ and $S(lwz)$, which takes $O((n_2+n_3)\log^2 n)$ time. Now modify the set $S(lxy)$ to construct the set $S(lwx)$, and then use the sets $S(lwx)$ and $S(lwz)$ to construct $S(wxz)$, which takes $O(n_3 \log^2 n)$ time. Observe that after the modification of the set $S(lxy)$, we are left with the set $S(wxy)$.

We now show that the total time taken is $T(n) \le dn \log^3 n$, for some constant d, as follows. There exists $c > 0$ such that for all $d \ge c$,

$$\begin{aligned}
T(n) &= T(n_1) + T(n_2) + T(n_3) + O((n_2+n_3)\log^2 n) \\
&\le dn_1 \log^3 n_1 + dn_2 \log^3 n_2 + dn_3 \log^3 n_3 + c(n_2+n_3)\log^2 n \\
&\le dn_1 \log^3 n + dn_2 \log^2 n \log \tfrac{n}{2} + dn_3 \log^2 n \log \tfrac{n}{2} + c(n_2+n_3)\log^2 n \\
&= dn_1 \log^3 n + dn_2 \log^2 n(\log n - 1) + dn_3 \log^2 n(\log n - 1) + c(n_2+n_3)\log^2 n \\
&= d(n_1+n_2+n_3)\log^3 n - (d-c)(n_2+n_3)\log^2 n \\
&\le dn \log^3 n.
\end{aligned}$$

Observe that the construction of the initial data structure $\mathcal{D}$ takes $O(n \log^2 n)$ time, which is dominated by $T(n)$. The dynamic planar convex hull of Brodal and Jacob [22] takes amortized $O(\log n)$ time per update. Therefore, using their data structure instead of Overmars and van Leeuwen's data structure [21] we can improve the expected running time of our algorithm. Since the algorithms of [21, 20] take linear space, the space complexity of our algorithm is $O(n)$.

Theorem 1. *Given a plane 3-tree G with n vertices and a set S of n points in general position in $\mathbb{R}^2$, we can decide the point-set embeddability of G on S in $O(n \log^3 n)$ time and $O(n)$ space, and compute such an embedding if it exists.*

Under the assumption that the algorithms of Overmars and van Leeuwen [21] and Steiger and Streinu [20] can handle degenerate cases, it is straightforward to modify our algorithm for the case when the input points are not necessarily in general position.

3 Universal Point Set for Plane 3-Trees

In this section we give an algorithm to compute 2-bend point-set embeddings of plane 3-trees on a set of n points in general position in $O(W^2)$ area, where W is the length of the side of the smallest axis-parallel square that encloses S.

We describe an outline of the algorithm. Given a plane 3-tree G and a set of points S in general position, we first construct a straight-line drawing Γ of G such that every point of S other than a pair of points on the convex hull of S lies in the proper interior of some distinct inner face in Γ, as shown in Figure 3(b). While constructing Γ, we compute a bijective function ϕ from the vertices of Γ to the points of S. We then extend each edge (u, v) in Γ using two bends to place the vertices u and v onto the points $\phi(u)$ and $\phi(v)$, respectively, as shown in Figure 3(c). We prove that Γ and ϕ maintain certain properties so that the resulting drawing Γ' remains planar.

In the following we describe the algorithm in detail. Let H be the convex hull of S. Construct a triangle xyz with $O(W^2)$ area such that xyz encloses H and the side yz passes through a pair of consecutive points y', z' on the boundary of H. Assume that y' is closer to y than z'. Set $\phi(y) = y'$ and $\phi(z) = z'$. Set $\phi(x) = x'$, where x' is the point on the convex hull of $S(xyz)$ for which the angle $\angle xyx'$ is smallest. Figure 3(a) illustrates the triangle xyz and the function ϕ. We call the straight line segments $x\phi(x), y\phi(y), z\phi(z)$ the *wings* of xyz. Observe that only $x\phi(x)$ among the three wings of xyz lie in the proper interior of xyz. We use this invariant throughout the algorithm, i.e., every face f in the drawing will contain at most one wing that is in the proper interior of f. We call such a wing the *major wing* of f.

Let a, b, c be the outer vertices of G in anticlockwise order and let p be the representative vertex of G. Map the vertices a, b, c to the points x, y, z. Let $S \setminus \{x', y', z'\}$ be the point set S'. Let n_1, n_2 and n_3 be the number of inner vertices of G_{abp}, G_{bcp} and G_{cap}, respectively. Since the major wing of xyz is incident to x, we construct a point $w \notin S$ such that $S'(wxy) = n_1, S'(wyz) =$

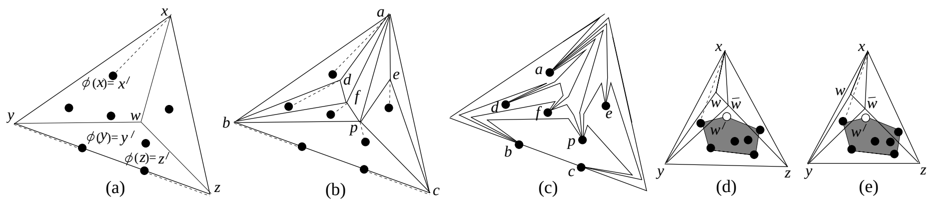

Fig. 3. (a) Illustration for the triangle xyz. (b) Γ and ϕ, where ϕ is illustrated with dashed lines. (c) A 2-bend point-set embedding of G on S. (d)–(e) Construction of w and $\phi(w)$, where $\phi(w) = w'$ is shown in white and the convex hull of $S(xyz)$ in gray.

$n_2 + 1$ and $S'(wxz) = n_3$, as shown in Figure 3(a). Steiger and Streinu [20] proved that such a point always exists and gave an $O(|S'|)$-time algorithm to find w. Since the angle $\angle xy\phi(x)$ is the smallest, if wy or wz intersects $x\phi(x)$, then by continuity there must exist another point $\bar{w}$ on the line wz such that $S'(\bar{w}xy) = n_1, S'(\bar{w}yz) = n_2 + 1$, $S'(\bar{w}xz) = n_3$ holds, and we choose $\bar{w}$ as the point w. Figures 3(d)–(e) depict such scenarios. Set $\phi(w) = w'$, where w' is the point on the convex hull of $S'(wyz)$ for which the angle $\angle wyw'$ is smallest. Since wyz does not contain $x\phi(x)$, the mapping we compute maintains the invariant that every face contains at most one major wing.

We now recursively construct the drawings of G_{abp}, G_{bcp} and G_{cap} with the point sets $S'(xyw)$, $S'(yzw)\backslash w'$ and $S'(zxw)$, respectively. Note that while recursively constructing a point w for the representative vertex inside some triangle xyz, then the triangle may not have any major wing. Also in this case, it suffices to compute w such that $S'(wxy) = n_1, S'(wyz) = n_2 + 1$ and $S'(wxz) = n_3$ holds. Once we complete the recursive computation, we obtain a straight-line drawing Γ of G, and a bijective function ϕ from the vertices of Γ to the points of S. We now extend each edge (u, v) in Γ using two bends to place the vertices u and v onto the points $\phi(u)$ and $\phi(v)$, respectively. We use ϕ and the property that every face in Γ contains at most one major wing, to maintain planarity. We omit the details due to space constraints.

Theorem 2. *Given a plane* 3*-tree* G *with* n *vertices and a point set* S *of* n *points in general position, we can compute a* 2*-bend point-set embedding of* G *in* $O(n \log^3 n)$ *time with* $O(W^2)$ *area, where* W *is the length of the side of the smallest axis-parallel square that encloses* S*.*

4 Approximate Point-Set Embeddings

Let Γ be a straight-line drawing of G. Then $S(\Gamma)$ denotes the number of vertices in Γ that are mapped to distinct points of S. The *optimal point-set embedding* of G is a straight-line drawing Γ^* such that $S(\Gamma^*) \geq S(\Gamma')$ for any straight-line drawing Γ' of G. A ρ-approximation point-set embedding algorithm computes a straight-line drawing Γ of G such that $S(\Gamma)/S(\Gamma^*) \geq \rho$. In this section we show that given a plane 3-tree G with n vertices, we can construct a straight-line drawing Γ of G such that $S(\Gamma) = \Omega(\sqrt{n})$, and hence point-set embeddability is approximable with factor $\Omega(1/\sqrt{n})$ for plane 3-trees.

Let G be a plane 3-tree with the outer vertices a, b, c and representative vertex p, and let the number of vertices of G be n. Then the *representative tree* T_{n-3} of G satisfies the following conditions [11].

(a) If $n = 3$, then T_{n-3} is empty.
(b) If $n = 4$, then T_{n-3} consists of a single vertex.
(c) If $n > 4$, then the root p of T_{n-3} is the representative vertex of G and the subtrees rooted at the three counter-clockwise ordered children p_1, p_2 and p_3 of p in T_{n-3} are the representative trees of G_{abp}, G_{bcp} and G_{cap}, respectively.

Since a rooted tree with n nodes is a partially ordered set under the 'successor' relation, by Dilworth's theorem [23], either the height or the number of leaves in the tree is at least $\sqrt{n}$. Let G be the input plane 3-tree with n vertices and let T be its representative tree with $n - 3$ vertices [11].

If T has $\Omega(\sqrt{n})$ leaves, then it is straightforward to construct a drawing Γ of G using the algorithm of Steiger and Streinu [20] such that exactly the leaves are mapped to the points of S, i.e., $S(\Gamma) = \Omega(\sqrt{n})$. Otherwise, the height of T is $\Omega(\sqrt{n})$. In this case we prove that G has a 'canonical ordering tree'(also, called Schnyder's realizer [2]) with height $\Omega(\sqrt{n})$, as shown in Lemma 4. There exists a simple algorithm (one can also modify de Fraysseix et al.'s algorithm [1]) to obtain a straight-line drawing Γ of G such that $S(\Gamma) = \Omega(\sqrt{n})$. We omit the details due to space constraints.

Lemma 4. *Let G be a plane 3-tree and let T be its representative tree. If the height of the representative tree is $\Omega(\sqrt{n})$, then G has a canonical ordering tree with height $\Omega(\sqrt{n})$.*

Proof. Let $P = (v_1, v_2, \ldots, v_k)$, $k = \Omega(\sqrt{n})$, be the longest path from the root v_1 of T to some leaf v_k. Without loss of generality assume that k is even. By G_i we denote the plane 3-tree induced by the outer vertices of G and the vertices $v_1, v_2, \ldots, v_k$. We now incrementally construct G_k. First construct a triangle xyz, place the vertex v_1 interior to xyz and add the segments v_1x, v_1y, v_1z. Since v_2 is a child of v_1, v_2 must be placed interior to one of the triangles incident to v_1. Since v_{i+1}, where $i + 1 \leq k$, is a child of v_i, this condition holds throughout the construction. Let T_x, T_y, T_z be the trees of the Schnyder's realizer rooted at x, y, z, respectively. Figure 4(a) illustrates the realizer of G_2, where the height of T_x, T_y and T_z is two, two and three, respectively. By A and B we denote the rooted trees isomorphic to T_x and T_z of G_2, respectively. The nodes of $T_w, w \in \{x, y, z\}$, where the realizer grows while adding v_{i+1} to G_i, $i \geq 2$, are called the *connectors* of T_w in G_i. See Figure 4(b).

Consider now the steps when we obtain the graphs $G_2, G_4, \ldots, G_k$. Observe that each time some tree of the form A (or B) gets connected with some $T_w, w \in \{x, y, z\}$, of G_i, the connectors of A (or B) become the only connectors of T_w in G_{i+2}. Figures 4(c)–(d) illustrate such a scenario. Consequently, each time some tree of the form B gets connected with some $T_w, w \in \{x, y, z\}$, of G_i, the height

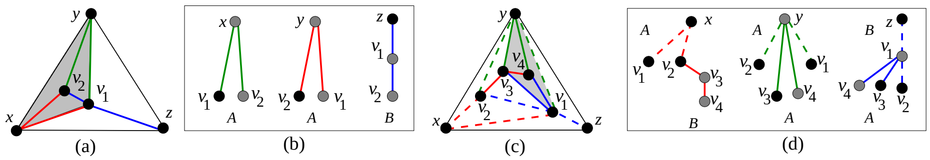

Fig. 4. (a) Illustration for G_2, where T_x, T_y and T_z are shown in red, green and blue. (b) Illustration for the connectors, shown in gray. (c)–(d) Example of a connection of A, A, B with B, A, A, respectively.

of T_w increases by one in G_{i+2}. Since we need to encounter $k/2$ steps before we obtain G_k, one of T_x, T_y or T_z must have height at least $k/6 = \Omega(\sqrt{n})$. Since each tree of the Schnyder's realizer of G_k is a subtree of a distinct tree of the Schnyder's realizer of G, the proof is complete. □

Theorem 3. *Given a plane 3-tree G with n vertices and a point set S of n points in general position in $\mathbb{R}^2$, we can compute a straight-line drawing Γ of G in polynomial time such that the number of vertices in Γ that are mapped to distinct points of S is $\Omega(1/\sqrt{n})$ times to the optimal. Hence the point-set embeddability of plane 3-trees is approximable with factor $\Omega(1/\sqrt{n})$.*

5 Conclusion

Using techniques that are completely different from those used in the previously best known approaches for testing point-set embeddability of plane 3-trees (achieving $O(n^{4/3+\varepsilon})$ time and $O(n^{4/3})$ space), in Section 2 we described an algorithm that solves the problem for a given plane 3-tree in $O(n \log^3 n)$ time using $O(n)$ space. As suggested by an anonymous reviewer, one possibility for potentially reducing the running time further might be to apply the algorithm of Moosa and Rahman [13], where an orthogonal range search would be used instead of a triangular range search. Specifically, given points x and y and an integer k, a triangle wxy that contains k points can be found by encoding each point w using two values: the slopes of wx and wy. The triangle wxy is then mapped to a two-sided axis-aligned orthogonal range query. It is not obvious, however, how this technique would be applied in recursive levels. One possibility might be to use a dynamic orthogonal range counting data structure. Another interesting issue is to examine the amount of scale up required to ensure that the vertices and bend points of the drawings produced in Section 3 lie on integer coordinates, i.e., the area requirement under minimum resolution assumption. In Section 4 we gave an $\Omega(1/\sqrt{n})$-approximation algorithm for plane 3-trees. Hence a natural question is to examine whether a constant factor approximation algorithm exists.

Acknowledgement. We thank Valentin Polishchuk and the other anonymous reviewers for many constructive and helpful comments.

References

1. de Fraysseix, H., Pach, J., Pollack, R.: How to draw a planar graph on a grid. Combinatorica **10**(1) (1990) 41–51
2. Schnyder, W.: Embedding planar graphs on the grid. In: SODA, ACM (1990) 138–148
3. Bose, P.: On embedding an outer-planar graph in a point set. Computational Geometry: Theory and Applications **23**(3) (2002) 303–312
4. Cabello, S.: Planar embeddability of the vertices of a graph using a fixed point set is NP-hard. Journal of Graph Algorithms and Applications **10**(2) (2006) 353–363
5. Ikebe, Y., Perles, M.A., Tamura, A., Tokunaga, S.: The rooted tree embedding problem into points in the plane. Discrete & Comp. Geometry **11**(1) (1994) 51–63
6. Bose, P., McAllister, M., Snoeyink, J.: Optimal algorithms to embed trees in a point set. Journal of Graph Algorithms and Applications **1**(2) (1997) 1–15
7. Castañeda, N., Urrutia, J.: Straight line embeddings of planar graphs on point sets. In: CCCG. (1996) 312–318
8. Durocher, S., Mondal, D.: On the hardness of point-set embeddability. In: WALCOM. Volume 7157 of LNCS., Springer (2012) 148–159
9. Biedl, T., Vatshelle, M.: The point-set embeddability problem for plane graphs. In: SoCG, ACM (2012) 41–50
10. Andrade, Jr., J.S., Herrmann, H.J., Andrade, R.F.S., da Silva, L.R.: Apollonian networks: Simultaneously scale-free, small world, euclidean, space filling, and with matching graphs. Physical Review Letters **94** (2005)
11. Nishat, R.I., Mondal, D., Rahman, M.S.: Point-set embeddings of plane 3-trees. Computational Geometry: Theory and Applications **45**(3) (2012) 88–98
12. Durocher, S., Mondal, D., Nishat, R.I., Rahman, M.S., Whitesides, S.: Embedding plane 3-trees in $\mathbb{R}^2$ and $\mathbb{R}^3$. In: GD. Volume 7034 of LNCS., Springer (2012)
13. Moosa, T.M., Rahman, M.S.: Improved algorithms for the point-set embeddability problem for plane 3-trees. In: COCOON. Volume 6842., Springer (2011) 204–212
14. Erickson, J.: On the relative complexities of some geometric problems. In: CCCG. (1995) 85–90
15. Brandenburg, F.J., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Liotta, G., Mutzel, P.: Selected open problems in graph drawing. In: GD. Volume 2912 of LNCS., Springer (2004) 515–539
16. Everett, H., Lazard, S., Liotta, G., Wismath, S.K.: Universal sets of n points for one-bend drawings of planar graphs with n vertices. Discrete & Computational Geometry **43**(2) (2010) 272–288
17. Kaufmann, M., Wiese, R.: Embedding vertices at points: Few bends suffice for planar graphs. J. of Graph Algorithms and Applications **6**(1) (2002) 115–129
18. Di Giacomo, E., Liotta, G.: The Hamiltonian augmentation problem and its applications to graph drawing. In: WALCOM. Volume 5942., Springer (2010) 35–46
19. Chazelle, B., Sharir, M., Welzl:, E.: Quasi-optimal upper bounds for simplex range searching and new zone theorems. Algorithmica **8**(5&6) (1992) 407–429
20. Steiger, W.L., Streinu, I.: Illumination by floodlights. Computational Geometry: Theory and Applications **10**(1) (1998) 57–70
21. Overmars, M.H., van Leeuwen, J.: Maintenance of configurations in the plane. Journal of Computer and System Sciences **23**(2) (1981) 166–204
22. Brodal, G.S., Jacob, R.: Dynamic planar convex hull. In: FOCS, IEEE (2002) 617–626
23. Dilworth, R.P.: A decomposition theorem for partially ordered sets. Annals of Mathematics **51**(1) (1950) 161–166