Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computational Geometry 29 (2004) 91–116
Computational Geometry
Theory and Applications
www.elsevier.com/locate/comgeo

Proximity drawings in polynomial area and volume ☆

Paolo Penna [a,∗,1], Paola Vocca [b]

[a] *Institut für Teoretische Informatik, ETH Zentrum, Clausiusstrasse 49, CH-8092 Zurich, Switzerland*
[b] *Dipartimento di Matematica, Università di Roma "Tor Vergata", Via della Ricerca Scientifica, I-00133 Rome, Italy*

Received 6 March 2001; received in revised form 17 May 2002; accepted 2 March 2004

Available online 18 May 2004

Communicated by K. Mehlhorn

Abstract

We introduce a novel technique for drawing proximity graphs in polynomial area and volume. Previously known algorithms produce representations whose size increases exponentially with the size of the graph. This holds even when we restrict ourselves to binary trees. Our method is quite general and yields the first algorithms to construct (a) *polynomial area weak Gabriel drawings of ternary trees*, (b) *polynomial area weak β-proximity drawing of binary trees* for any $0 \leqslant \beta < \infty$, and (c) *polynomial volume weak Gabriel drawings of unbounded degree* trees. Notice that, in general, the above graphs *do not admit a strong proximity drawing.* Finally, we give evidence of the effectiveness of our technique by showing that a class of graph requiring *exponential area* even for weak Gabriel drawings, admits a *linear-volume strong β-proximity drawing* and a *relative neighborhood drawing.* All described algorithms run in *linear time.*

Keywords: Graph drawing; Proximity; 3-dimensional drawing; Area-requirement

1. Introduction

A *proximity graph* is a geometric graph where a given set of points represents the vertices and two vertices are adjacent if and only if they are *neighbors* according to some definition *of neighborhood*. For example, the *Gabriel graph* of a set of points [23,33] is obtained by connecting every two points u and v

☆ A preliminary version of this work appeared in Proc. of GD'98 and Proc. of CCCG'98. Work partially supported by the ESPRIT Basic Research Action on "Algorithms and Complexity in Information Technology" (ALCOM-IT).

∗ Corresponding author.
E-mail addresses: penna@inf.ethz.ch (P. Penna), vocca@mat.uniroma2.it (P. Vocca).

[1] Work done while at the university of Rome "Tor Vergata", Math. Department.

0925-7721/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comgeo.2004.03.015

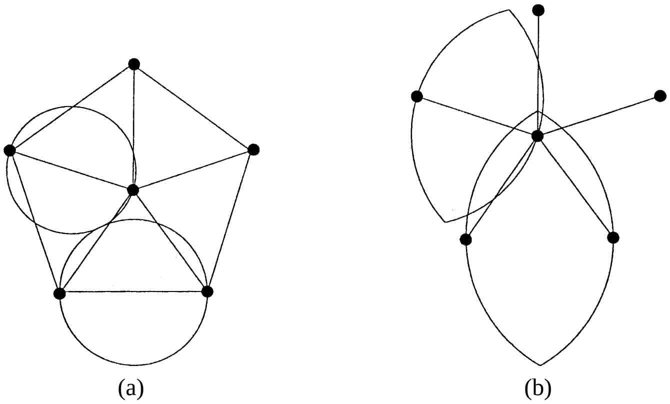

Fig. 1. The proximity graph of a point set changes when different proximity regions are considered: (a) a strong Gabriel graph; (b) a drawing which is both a strong 2-proximity drawing and a relative neighborhood graph.

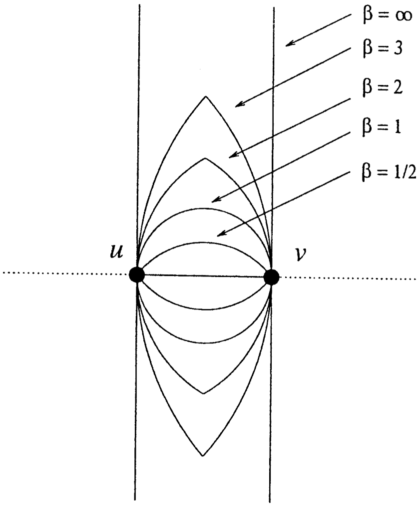

Fig. 2. β-proximity regions for $\beta = 1/2, 1, 2, 3, \infty$.

such that the closed disk having u and v as antipodal points does not contain any other point (see the example in Fig. 1(a)). Notice that, to a given set of points corresponds a unique graph whose vertices are the points on the plane and edges are determined by the positions of the vertices.

A natural extension of Gabriel graphs consists of defining a suitable *proximity region* of the vertices which determines the set of edges as follows: Two vertices are adjacent if and only if the corresponding proximity region is *empty*, i.e., it does not contain any other vertex of the graph.

In particular, in *β-proximity graphs* the proximity region (*β-region*) is a suitable lune depending on the parameter β, as shown in Fig. 2 (see Section 1.3 for a formal definition). In Fig. 1(b) we show the β-proximity graph for $\beta = 2$ and for the same set of points in Fig. 1(a). Clearly, for different values of β, the same set of points may yield different graphs. Variants in which open or closed lunes can be also considered. For instance, *relative neighborhood graphs* (RNG) are proximity graphs where the proximity region is the *open* lune of parameter $\beta = 2$.

This and other kind of proximity graphs have been deeply investigated due to the many applications in computational morphology, geographic information systems, pattern recognition and classification, computational geometry, and computer vision (see e.g. [23,27,33,38,41,42]).

Because of such applications, one of the most fundamental problem is that of *characterizing* the class of proximity graphs for a given definition of proximity. From the algorithmic point of view, the above question corresponds to decide whether a given graph can be realized as a proximity graph (e.g., is there a set of points S such that the Gabriel graph $GG(S)$ is isomorphic to the given graph?). Clearly, it would be extremely helpful for the applications to visualize the proximity graph, if any. This requires the computation of a *proximity drawing*, that is, a geometric representation of the input graph as a proximity graph. (See Section 1.3 for a formal definition of β-drawing.)

In general, constructing a "nice" drawing of a given graph is a per se very interesting problem since the drawing has to be displayed on a physical device with finite resolution. This imposes a finite resolution on the drawing as well (e.g., any two vertices must be at distance at least one) and also imposes the size of the drawing (e.g., the area of the smallest rectangle containing it) to be polynomially bounded in the size of the input graphs.

Therefore, the construction of a proximity drawing can be considered a very challenging problem since the drawing has to simultaneously satisfy the proximity constraints and some of the "classical" constraints of *graph drawing* (see the book [15] for an overview). In particular, the ability to construct area/volume-efficient drawings is essential in practical visualization applications, where saving screen space is of utmost importance. This property is meaningful only if the adopted drawing conventions prevent drawings from being arbitrarily scaled down. This is usually accomplished by assuming a *vertex resolution rule*, i.e., any two vertices must have distance at least one. For example, *grid drawings* satisfy the vertex resolution rule in that they impose vertices to have integer coordinates.

1.1. Previous related work

Unfortunately, it is quite difficult to characterize proximity graphs. For instance, no characterization of Gabriel graphs is known so far. Therefore, the research has been focused on the problem of constructing proximity drawings of certain classes of graphs. In [31] the drawability of outerplanar graphs as RNGs has been proved, while in [28] this result has been extended to β-proximity drawings.

Another well studied class of graphs for proximity drawability is that of trees. Although every tree is a subgraph of a maximal outerplanar graph, the positive results in [28,31] do not apply to trees as the characterizations of those trees that admit proximity drawings given in [3,4] show.

Motivated by the fact that several interesting classes of graphs do not admit a proximity drawing, the notion of *weak proximity* have been first introduced in [17]. Informally, a *weak proximity drawing* is a straight-line drawing such that, for any edge (u, v), the proximity region of u and v is empty. This definition relaxes the requirement of classical β-drawings, allowing the β-region of non-adjacent vertices to be empty. Classical, not weak, proximity drawings are generally referred to as *strong* proximity drawings. Interestingly, this simple modification allows for much more flexibility and efficacy. For instance, a tree that has a vertex of degree greater than five has no (strong) β-drawing for any β, while it admits a weak β-proximity drawing [17].

Another way of extending the class of drawable graphs is to consider 3-*dimensional proximity drawings*. In the 3-dimensional space the definition of β-proximity is the natural extension in which proximity regions are defined as intersections of spheres (e.g., the Gabriel proximity region is a *Gabriel* *sphere* instead of disk). Three-dimensional β-proximity drawings have been investigated in [29] where characterizations of drawable trees have been presented.

Other results on algorithms to construct proximity drawings of graphs and some related issues can be found in [22,35] and [21] (see also [16] for a good survey on proximity drawability).

More generally, algorithms for *graph drawing* have been extensively studied for a number of aesthetic criteria (e.g., planar drawings) and optimization functions (e.g., the area of the drawing) depending on the applications at hand (see [14,15] for an overview). For instance, rooted trees can be represented using *upward straight-line planar drawings* so to emphasize their hierarchical structure: (a) vertices are represented as points and no vertex can be placed above its parent; (b) each edge is represented as a straight-line segment connecting its endpoints; and (c) no two edges cross.

Optimal-area algorithms for drawing trees according to the above criteria have been investigated in several works [8,11–13,19,34]. Variants in which edges are represented as polylines (i.e., chains of segments connecting the endpoints) [8,24], vertices must be represented as boxes of given sizes [20, 37], or the *aspect ratio* (see Section 1.3 for a formal definition) has to be optimized [8], have been also considered. Other classes of graphs for upward drawing have been studied in [18,25,40], Finally, motivated by the availability of low-cost workstations and applications requiring three-dimensional representations of graphs [5,26,32,36,39], the construction of *three-dimensional drawings* of *polynomial volume* has been investigated in [1,2,6,7,9,10].

It is worth observing that many of the above cited works present algorithms yielding area/volume-efficient grid drawings. For instance, binary trees and bounded degree search trees[2] admit $\Theta(n \log n)$- and $\Theta(n)$-area algorithms for upward drawing, respectively [11,12,40]. Also, if we relax the upward requirement or we allow polylines to represent edges, than any binary tree admits a linear-area drawing [24,43]. Similar positive results have been also achieved for three-dimensional drawings (see e.g. [7]).

On the contrary, all known algorithms that compute both strong and weak proximity drawings produce representations whose area/volume *increases exponentially* with the number of vertices [3,4,16,17,21, 22,29–31,33]. This holds even when we restrict ourselves to binary trees and to any vertex resolution rule (instead of the more restrictive grid drawings). Indeed, the problem of constructing proximity drawings of graphs that have small size is considered a very challenging one by several authors [4,21]. Additionally, in [30] an *exponential lower bound* on the area of Gabriel drawings (both weak and strong) has been presented. Hence, the research in this field focused on characterizing classes of graphs that admit polynomial-size drawings.

1.2. Our contribution

In this paper, we introduce a general framework for drawing proximity graphs in polynomial area/volume, which starting from a suitable drawing Δ (not a proximity drawing), transforms Δ into a weak proximity drawing Δ'. The drawing Δ can be either 2- and 3-dimensional, and the area/volume of the final drawing Δ' is polynomially related to the area/volume of Δ. Up to our knowledge, this is the first algorithmic technique for polynomial-size proximity drawing.

The technique is general enough to be applied to a wide class of weak Gabriel drawable graphs. In particular, we first apply it to 2-dimensional and then to 3-dimensional drawings of trees with n vertices

[2] The definition of search tree used in [12] includes k-balanced, red-black and BB[α]-trees.

and, finally, to the class of planar triangular graphs G_n used in [30] to prove the exponential lower bound on the area of any strong (weak) proximity drawing (see Section 4 for a formal definition of G_n). As a result we obtain the first algorithms to construct polynomial-size β-proximity drawings for non trivial classes of graphs. In the sequel we list our results:

- A linear-time $n^2/2$-area algorithm for (upward) weak Gabriel drawing of ternary (rooted) trees using integer coordinates and constant aspect ratio;
- A linear-time $O(n^2)$-area algorithm for (upward) weak β-drawing of binary (rooted) trees, for $0 \leqslant \beta < \infty$, using integer coordinates and constant aspect ratio;
- A linear-time polynomial-volume algorithm for (strictly-upward) 3-dimensional weak Gabriel drawing of unbounded degree (rooted) trees, where the coordinates of vertices can be represented with $O(\log n)$-bits;
- A linear-time and linear-volume strong β-drawing, for $1 \leqslant \beta < 2$, and relative neighborhood drawing (RND) of the class of graphs G_n, where the coordinates of vertices can be represented with $O(\log n)$-bits.

Notice that, in the two dimensional case we use integer coordinates to represent vertices (i.e., grid drawing), while the three-dimensional drawings use coordinates which can be represented using $\Theta(\log n)$ bits. Indeed, the vertex resolution rule implies a lower bound of $\Omega(\log n)$ bits since we need to represent a set of n distinct-points. So, $O(\log n)$ bit-requirement is an important feature for an efficient representation.

In Table 1 we compare our results with the previously known results for the same class of graphs we consider in this work. Besides the fact that all previously known algorithms yield exponential area/volume drawings, our algorithms produce weak proximity β-drawings for classes of graphs that *do not admit strong β-proximity drawings*, at least for some β, Moreover, for the only case in which the graphs admit

Table 1
Our results versus previously known results on the existence of weak/strong β-proximity drawings (whenever not specified, previous results refer to two-dimensional drawings and/or to the same value of β as in our results)

Class	Our results			Previous results
	Size	β	Weak/strong	Drawability
Ternary trees	$n^2/2$-area	$0 \leqslant \beta \leqslant 1$	Weak	Not strong [16]
Binary trees	$O(n^2)$-area	$0 \leqslant \beta < \infty$	Weak	Not strong for $0 \leqslant \beta \leqslant \frac{\sqrt{3}}{2}$ [3], strong for $\frac{\sqrt{3}}{2} < \beta \leqslant \infty$ [3]
Unbounded degree trees	$O(n^4)$-volume	$0 \leqslant \beta \leqslant 1$	Weak	Not strong [29] (even in 3D)
G_n	$O(n)$-volume	$1 \leqslant \beta < 2$, RND	Strong	Strong for $\beta \leqslant \frac{1}{1-\cos 2\pi/5}$ [33] $\Omega(3^n)$-area [30] (also for weak)

strong proximity drawings (i.e., the graphs G_n introduced in [33]) our method also yields polynomial-size strong proximity drawings.

Finally, the importance of our result on the class G_n is twofold. First, it shows that our method is general enough to be applied to classes of graphs other than trees. Second, the class G_n exhibit an exponential gap between the area and volume requirement. By one hand, in [30] an exponential lower bound on the area, even when restricted to weak proximity drawings, has been proved. By the other hand, our technique yields a linear-volume strong proximity drawings. This results shows how the use of the third dimension can substantially help in improving the efficiency of the proximity drawings.

Paper organization. In Section 1.3, we recall basic definitions and introduce the notation adopted. In Section 2 we describe the drawing framework and state its main properties. In Section 3 and in Section 4 we apply our technique to 2-dimensional and 3-dimensional drawings, respectively. Finally, in Section 5, future research directions are outlined.

1.3. Preliminaries and notation

Given a pair of points in the plane u and v, let $d(u,v)$ denote the Euclidean distance. The proximity region of u and v, also referred to as *β-region of influence of u and v*, denoted by $R[u,v,\beta]$, is defined as follows (see also Fig. 2):

(1) For $0 < \beta < 1$, $R[u,v,\beta]$ is the intersection of the two closed disks of radius $d(u,v)/(2\beta)$ passing through both u and v.
(2) For $1 \leqslant \beta < \infty$, $R[u,v,\beta]$ is the intersection of the two closed disks of radius $\beta d(u,v)/2$ and centered at the points $(1-\beta/2)u + (\beta/2)v$ and $(\beta/2)u + (1-\beta/2)v$.
(3) For $\beta = 0$, $R[u,v,0]$ is the segment having u and v as endpoints.
(4) $R[u,v,\infty]$ is the closed infinite strip perpendicular to the line segment $\overline{uv}$.

A *weak β-drawing* of a graph G is a planar straight-line drawing of G such that, for any two adjacent vertices u and v, the proximity region $R[u,v,\beta]$ does not contain any other vertex of the drawing.[3] If the proximity region of any two *non-adjacent* vertices contains at least another vertex then the drawing of G is a *strong β-drawing* or simply *β-drawing* (see the example in Fig. 1(b)).

A (weak) *Gabriel drawing* is a (weak) β-drawing for $\beta = 1$. In this case, the proximity region of any two points u and v is denoted as $R[u,v]$ and it corresponds to the closed disk of radius is $d(u,v)$ and centered at the middle point between u and v.

Similarly, we define β-proximity regions of 3-dimensional drawings as the intersection of closed spheres.

A graph G with n vertices is (*weak*) *β-drawable* if it admits a (weak) β-drawing (either 2-dimensional or 3-dimensional).

In the 2-dimensional space, a *layer l_i* is a horizontal line containing the points having y-coordinates equal to Y_i, where Y_i is a positive integer. Similarly, in the 3-dimensional space, a *layer l_i* is the plane containing the points having the z-coordinate equal to a positive integer Z_i. In the following we assume that $Y_{i+1} \geqslant Y_i$ and $Z_{i+1} \geqslant Z_i$, for any $i \geqslant 1$.

[3] To simplify the notation, we denote a vertex and a point representing it with the same symbol.

A *layered drawing* in a straight-line drawing such that each vertex is placed on a layer. Notice that, in this definition vertices on a same layer can be adjacent, and we allow layers not to be equally spaced. The number of layers of a layered drawing Δ is denoted as h^{Δ}.

Given a vertex u we denote by L_u the layer on which the vertex is drawn and, for any vertex v, v_u denotes the projection of v on layer L_u. Moreover, we define

$$R_u[u,v] \stackrel{\triangle}{=} R[u,v_u], \qquad d_u(u,v) \stackrel{\triangle}{=} d(u,v_u),$$

and for any layer L containing at least one vertex

$$d(L) \stackrel{\triangle}{=} \text{argmax}\{d_u(u,v) \mid u \in L \wedge v \text{ adjacent to } u\}.$$

To simplify the notation use d_i as a shorthand for $d(l_i)$. We also use d_i^{Δ} to denote $d(l_i)$ restricted to vertices that are adjacent in a subdrawing Δ, only.

As previously stated, in order to prevent drawings from being arbitrarily scaled down, we assume the *vertex resolution* rule, i.e., for any two distinct vertices u and v it must hold $d(u,v) \geqslant 1$. The *bit-requirement* is the number of bits needed to represent the coordinates of the vertices.

The *height*, the *width* and the *area* of a 2-dimensional drawing are the height, the width and the area of the smallest isothetic rectangle bounding the drawing, respectively. Analogously, the *height*, the *width*, the *depth* and the *volume* of a 3-dimensional drawing are defined as the height, the width, the depth and the volume, respectively, of the smallest isothetic parallelepiped bounding the drawing. The *aspect ratio* is defined as the ratio between the length of the longest side and the length of the shortest side of the smallest rectangle (parallelepiped, in the 3-dimensional case) containing the drawing.

Let u, v and z be any three points. We denote by $\triangle(uvz)$ the triangle whose vertices are u, v and z; $\angle uzv$ denotes the angle determined by the two segment lines $\overline{uz}$ and $\overline{vz}$ and whose value is in $[0,\pi]$.

2. The technique

In this section we introduce a framework for weak β-proximity drawing in polynomial area/volume, for any $0 \leqslant \beta \leqslant 1$. Since every weak Gabriel drawing is also a weak β-drawing, for $\beta \leqslant 1$, we will present the technique for Gabriel drawings (i.e., $\beta = 1$).

In particular, our method consists of two main steps: (a) construct a suitable (not Gabriel) drawing Δ; (b) transform Δ into a weak Gabriel drawing Δ'. The initial drawing Δ, titled *quasi-Gabriel drawing*, can be both 2- and 3-dimensional and the size (area/volume) of Δ' is polynomially bounded in the size of Δ. Hence, if a graph admits a quasi-Gabriel drawing of polynomial size, then the resulting weak Gabriel drawing is of polynomial size as well.

In the following, we first formally define a quasi-Gabriel drawing Δ and then we describe the transformation of Δ into a weak Gabriel drawing Δ'.

Definition 2.1. A drawing Δ is a *quasi-Gabriel drawing* if the following constraints hold:

(1) *Layered*. Vertices lie on layers;
(2) *No Transitive Edges*. Vertices on non-consecutive layers are not adjacent;
(3) *Locally Gabriel*. For any edge (u,v), $R_u[u,v] \cap L_u$ contains no vertices other than u and v.

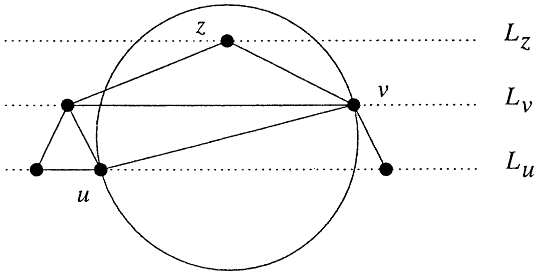

Fig. 3. A quasi-Gabriel drawing.

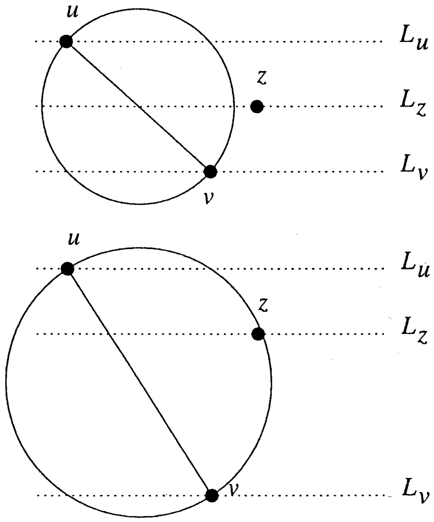

Fig. 4. Layered drawings with transitive edges cannot be "stretched" without introducing new vertices in a proximity region that was originally empty.

Fig. 3 shows an example of a quasi-Gabriel drawing: notice that the vertex z is contained in $R[u, v]$, thus, not satisfying the definition of weak Gabriel drawing. However, the drawing can be easily adjusted by increasing the distance between L_z and L_u so that L_z does not intersect $R[u, v]$ anymore. In general, increasing the distance between layers makes some proximity region bigger and may introduce a new vertex in a region that was originally empty: Fig. 4 shows an example of a layered drawing which *does not* satisfy the "No Transitive Edges" property of Definition 2.1. In the sequel we will show that this problem cannot occur in a quasi-Gabriel drawing.

Informally speaking, our technique is based on the following ideas:

(1) In the starting quasi-Gabriel drawing, if $R[u, v]$ contains another vertex z, then z cannot lie on L_u nor on L_v.
(2) After spacing out consecutive layers by a suitable amount, every proximity region $R[u, v]$ intersects L_u and L_v only. Therefore, in the new drawing $z \notin R[u, v]$.

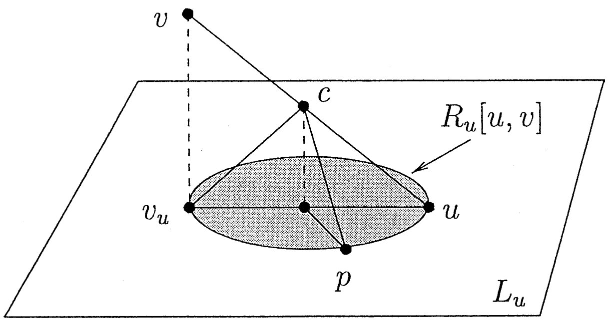

Fig. 5. The proof of Lemma 2.2.

(3) Although increasing the distance between two consecutive layers L_u and L_v makes the proximity region $R[u,v]$ bigger, the intersection of $R[u,v]$ with L_u and L_v does not change. This implies that we never introduce new vertices while enlarging $R[u,v]$.

The following lemma easily implies that, for any two adjacent vertices u and v in a quasi-Gabriel drawing, no other vertex $z \in L_u \cup L_v$ is contained in $R[u,v]$.

Lemma 2.2. *For any two vertices u and v it holds that*

$$R[u,v] \cap L_u = R_u[u,v] \cap L_u.$$

Proof. In the 2-dimensional case we simply observe that both $R[u,v] \cap L_u$ and $R_u[u,v] \cap L_u$ coincides with the segment having u and v as endpoints.

As for the 3-dimensional case, we first observe that $R_u[u,v] \cap L_u$ is the closed disk on L_u of endpoints u and v_u (see Fig. 5). Indeed, $R_u[u,v]$ is a sphere whose center c_u lie on L_u and whose diameter equals $d(u,v_u)$. In order to prove the lemma, we will show that, for any point $p \in L_u$, it holds that

$$d(c,p) \leqslant d(u,v)/2 \quad \Leftrightarrow \quad p \in R_u[u,v] \cap L_u, \tag{1}$$

where c is the center of $R[u,v]$. Towards this aim, we consider the two triangles $\triangle(u,c,c_u)$ and $\triangle(p,c,c_u)$. As they have a common segment $\overline{cc_u}$ and $\angle cc_u v = \angle cc_u p = \pi/2$, it holds that

$$d(c,p) \leqslant d(c,u) \quad \Leftrightarrow \quad d(p,c_u) \leqslant d(u,c_u).$$

Since $d(c,u) = d(u,v)/2$ and $d(u,c_u) = d(u,v_u)/2$, the above condition is equivalent to Eq. (1). This completes the proof. □

The next lemma specifies how much the distance between layers should be increased.

Lemma 2.3. *Let u and v be any two adjacent vertices of a layered drawing and let L be a layer whose distance from both L_u and L_v is bigger than* $\max\{d(L_u), d(L_v)\}/2$. *Then, it holds that* $R[u,v] \cap L = \emptyset$.

Proof. Without loss of generality, let us suppose that layer L is closer to L_u than to L_v and let c be the center of the region of influence $R[u,v]$ (see Fig. 6). Also let c_u and c_L be the projection of c on layer L_u

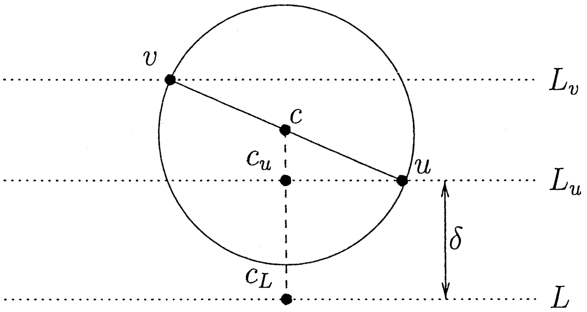

Fig. 6. The proof of Lemma 2.3.

and L, respectively. Since $d(c_u, c_L) = \delta > d(L_u)/2 \geqslant d_u(u, v)/2 = d(c_u, v)$, then the distance between c and L is equal to

$$d(c, c_L) = d(c, c_u) + d(c_u, c_L) \geqslant d(c, c_u) + d(c_u, v) > d(c, v).$$

Hence the lemma follows. □

We are now in a position to prove the main result of this section. The following theorem evaluates the dimensions of a weak Gabriel drawing Δ' derived from a quasi-Gabriel drawing Δ.

Theorem 2.4 (Drawing stretching). *Let Δ be a quasi-Gabriel* (*grid*) *drawing. A weak Gabriel* (*grid*) *drawing Δ' exists such that*:

- $\text{width}(\Delta') = \text{width}(\Delta)$;
- $\text{depth}(\Delta') = \text{depth}(\Delta)$;
- $\text{height}(\Delta') < 2\sum_{i=1}^{h^\Delta} \lfloor d_i/2 \rfloor + 1$.

Moreover, if $d_i \geqslant d_{i-1}$, for $2 \leqslant i \leqslant h^\Delta$, then $\text{height}(\Delta') < \sum_{i=1}^{h^\Delta} \lfloor d_i/2 \rfloor + 1$.

Proof. We construct Δ' by increasing the distance between consecutive layers of Δ. In particular, let us denote by δ_i the distance between layer l_i and layer l_{i-1} in Δ', for $2 \leqslant i \leqslant h^\Delta$. We set

$$\delta_i = \max\bigl\{\lfloor d_{i-1}/2 \rfloor + 1, \lfloor d_i/2 \rfloor + 1\bigr\}.$$

Thus

$$\text{height}(\Delta') \leqslant \sum_{i=2}^{h^\Delta} \delta_i < 2\sum_{i=1}^{h^\Delta} \lfloor d_i/2 \rfloor + 1.$$

Moreover, if $d_i \geqslant d_{i-1}$ for $2 \leqslant i \leqslant h^\Delta$, then

$$\text{height}(\Delta') = \sum_{i=2}^{h^\Delta} \lfloor d_i/2 \rfloor + 1.$$

In order to prove that Δ' is a weak Gabriel drawing we show that the region of influence $R[u, v]$ of any two adjacent vertices does not contain any other vertex z. We distinguish the following two cases:

- $z \in L_u \cup L_v$. Without loss of generality, we can assume $z \in L_u$. We first observe that Δ' is also a quasi-Gabriel drawing since the "Locally Gabriel" property is preserved: $R_u[u, v]$ does not change when increasing the distance between layers since the projection of v on L_u does not change. Therefore, the fact that Δ was a quasi-Gabriel drawing implies $z \notin R_u[u, v]$. Finally, Lemma 2.2 implies that $z \notin R[u, v]$.
- $z \notin L_u \cup L_v$. Let δ be the distance between L_z and the nearest between L_u and L_v. By construction, it holds that

  $$\delta \geqslant d(L_u)/2 \quad \text{and} \quad \delta \geqslant d(L_v)/2.$$

  Thus by applying Lemma 2.3 we have that $z \notin R[u, v]$.

Finally, if Δ is a grid drawing, then Δ' is a grid drawing as well. □

Let us observe that if Δ is a polynomial area/volume quasi-Gabriel drawing then the area/volume of Δ' is polynomial as well. Indeed, width(Δ') = width(Δ), depth(Δ') = depth(Δ), and height(Δ') is at most n-times (the maximum number of layers) the maximum between the width(Δ) and depth(Δ). Hence, the above theorem implies that classes of graphs that admit polynomial area/volume quasi-Gabriel drawings, also admit polynomial area/volume weak Gabriel drawings.

3. Proximity drawings in the plane

This section is devoted to the construction of upward proximity drawings in the plane for rooted trees. In particular, we will first prove that ternary trees admit $n^2/2$-area weak Gabriel grid drawings. Then, we will consider β-proximity grid drawings of binary trees, for $0 \leqslant \beta < \infty$. Notice that ternary trees do not admit *strong* Gabriel drawings, and binary trees are not strong β-drawable for $0 \leqslant \beta \leqslant \sqrt{3}/2$ (see Table 1).

3.1. Ternary trees

We apply the method described in Section 2 by showing how to construct a quasi-Gabriel drawing Δ of polynomial area.

For any ternary tree T two different drawings Δ^l and Δ^r are constructed. Let T_1, T_2 and T_3 be the ternary trees rooted at the children of the root of T such that T_1 and T_3 are the smallest and the largest one, respectively (ties are solved arbitrarily). We denote with Δ^l and Δ^r the two drawings of T recursively obtained by combining the drawings of T_1, T_2 and T_3, as shown in Fig. 7. The compositions of the three subdrawings used to obtain Δ^l and Δ^r are denoted as $\Delta_1^r \ominus \Delta_2^l \ominus \Delta_3^l$ and $\Delta_3^r \ominus \Delta_2^r \ominus \Delta_1^l$, respectively. In particular, Δ^l is obtained by translating both Δ_1^r and Δ_2^l by one unit to the bottom with respect to Δ_3^l. Moreover, the bounding box of Δ_1^r, Δ_2^l and Δ_3^l are pairwise at horizontal unit distance. Finally, the root of T is drawn on the same layer of the root of T_3 in Δ_3^l and its x-coordinate is an integer value strictly in between the x-coordinates of the roots of T_1 in Δ_1^r and of T_2 in Δ_2^l. Notice that this implies that $\angle r_1 r_2 < \pi/2$ and $\angle r_2 r_1 < \pi/2$, that is $r_2 \notin R[r, r_1]$ and $r_1 \notin R[r, r_2]$. We similarly define $\Delta_3^r \ominus \Delta_2^r \ominus \Delta_1^l$.

Algorithm `ternary-trees` in Fig. 8 constructs the quasi-Gabriel drawings Δ^l and Δ^r satisfying the following invariants:

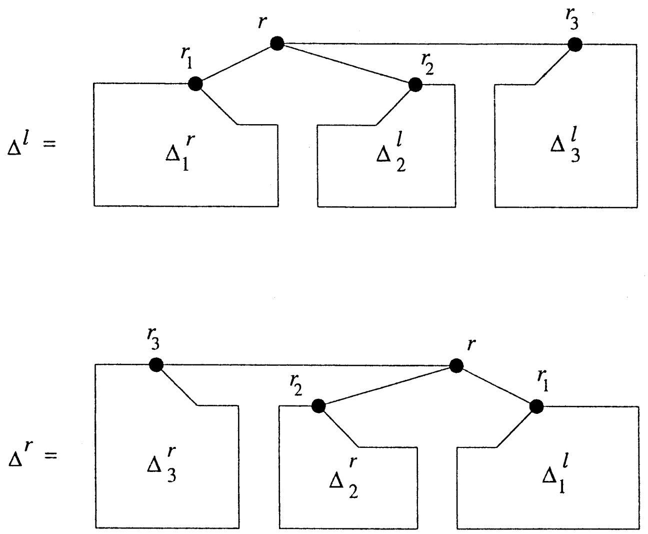

Fig. 7. The construction of the quasi-Gabriel drawings Δ^l and Δ^r for ternary trees.

algorithm `ternary-trees`(T)
$h \leftarrow$ height of T
$r \leftarrow$ root of T
if $h = 1$ **then**
  draw r at $(1, 1)$
  $\Delta^l, \Delta^r \leftarrow$ drawing of r
else begin
  $(\Delta_1^l, \Delta_1^r) =$ `ternary-trees`(T_1)
  $(\Delta_2^l, \Delta_2^r) =$ `ternary-trees`(T_2)
  $(\Delta_3^l, \Delta_3^r) =$ `ternary-trees`(T_3)
  $\Delta^l = \Delta_1^r \ominus \Delta_2^l \ominus \Delta_3^l$
  $\Delta^r = \Delta_3^r \ominus \Delta_2^r \ominus \Delta_1^l$
  end
return (Δ^l, Δ^r)
end

Fig. 8. Algorithm ternary-trees.

(1) Edges from a vertex to its children are represented with one horizontal, one downward leftward and one downward rightward line.
(2) The root is the leftmost vertex in Δ^l (rightmost in Δ^r, respectively) on the top layer.

Theorem 3.1. *Δ^l and Δ^r are quasi-Gabriel grid drawings.*

Proof. Let us consider the drawing Δ^l (the proof for Δ^r is similar and therefore omitted). It is easy to see that Δ^l is a layered drawing with no transitive edges. Thus, we have to prove that for any edge (u, v),

$R_u(u,v) \cap L_u$ does not contain any vertex other than u and v. The proof is by induction on the number of vertices n of the tree.

Base step ($n=1$). Trivial.

Inductive step. We distinguish the following two subcases.

- $v=r$. In this case $u=r_i$, for some $1 \leqslant i \leqslant 3$. Suppose $u=r_1$ (the other two cases are similar). Notice that (r,r_1) is represented as a downward leftward segment. Consider that, by construction:
  (1) r_1 is the rightmost vertex of Δ_1^r and on layer L_{r_1};
  (2) r_2 is the leftmost vertex of Δ_2^l on layer $L_{r_1} = L_{r_2}$;
  (3) r is drawn strictly in between the x-coordinates of r_1 and of r_2.
  Hence, $R_{r_1}[r,r_1] \cap L_{r_1}$ does not contain any vertex of Δ_1^r and Δ_2^l. By construction, it also contains no vertex of Δ_3^l.
- $u=r$. In this case $v=r_i$, for some $i \in \{1,2,3\}$. It easy to verify that $R_r[r,r_i]$ is empty for $i=1,2,3$.
- $u, v \neq r$. Without loss of generality, we assume that u, v are vertices of Δ_1^r. By inductive hypothesis, no other vertex of Δ_1^r belongs to $R_u[u,v] \cap L_u$. It is also easy to see that $R_u[u,v]$ is contained in the bounding box of Δ_1^r, thus implying that $R_u[u,v] \cap L_u$ does not contain any vertex other than u and v.

Finally, by construction, every vertex is represented as a point with integer coordinates. □

Lemma 3.2. *Let Δ be either Δ^l or Δ^r. For any $1 \leqslant i \leqslant h^\Delta$, $d_i \leqslant n/2^{h^\Delta - i+1}$.*

Proof. Without loss of generality, we assume $\Delta = \Delta^l$. The proof proceeds by induction on n. Let us denote with n_1, n_2 and n_3 the number of nodes of the three immediate subtrees, and let us suppose $n_1 \leqslant n_2 \leqslant n_3$.

Base step ($n=4$). Let us first consider the drawing of the complete ternary tree of height 2. In this case we clearly have $h^\Delta = 2$ and $d_2 = 2$. Moreover, it is easy to see that any other tree with 4 vertices admits a drawing Δ satisfying $d_i \leqslant n/2^{h^\Delta - i+1}$.

Inductive step. We distinguish the following two cases:

- $i = h^\Delta$. By definition, d_{h^Δ} is the length of the longest projection on layer h^Δ of any edge among $(r,r_1), (r,r_2), (r,r_3)$ and an edge on layer h^{Δ_3} of Δ_3^l (see Fig. 7). By inductive hypothesis and considering that T_3 is the largest subtree we have, $d_{h^\Delta} \leqslant n/2$.
- $2 \leqslant i \leqslant h^\Delta - 1$. Observe that, by construction (see Fig. 7), layer l_i of Δ corresponds to layer l_{i_j} in Δ_j, where

  $$i_1 = i - h^\Delta + h^{\Delta_1} + 1;$$
  $$i_2 = i - h^\Delta + h^{\Delta_2} + 1;$$
  $$i_3 = i - h^\Delta + h^{\Delta_3}.$$

  Therefore, by inductive hypothesis we have

  $$\begin{aligned} d_i^\Delta &= \max\left\{d^{\Delta_1}_{i-h^\Delta+h^{\Delta_1}+1}, d^{\Delta_2}_{i-h^\Delta+h^{\Delta_2}+1}, d^{\Delta_3}_{i-h^\Delta+h^{\Delta_3}}\right\} \\ &\leqslant \max\left\{n_1/2^{h^\Delta-i}, n_2/2^{h^\Delta-i}, n_3/2^{h^\Delta-i-1}\right\} < n/2^{h^\Delta-i+1}, \end{aligned}$$

  where the last inequality comes from $n_1 < n/2$ and $n_2 < n/2$. □

By combining Lemma 3.2, Theorem 3.1 and Theorem 2.4 and we can state the following result.

Corollary 3.3. *Any ternary tree with n nodes admits a $n^2/2$-area weak Gabriel grid drawing which can be constructed in $\mathrm{O}(n)$ time.*

Proof. The width of the quasi-Gabriel drawing Δ derived by algorithm `ternary-trees` in Fig. 8 is at most n. Hence, the weak Gabriel drawing Δ' has:

- width$(\Delta') \leqslant n$;
- height$(\Delta') < \sum_{i=1}^{h^\Delta} \lfloor d_i/2 \rfloor + 1 \leqslant \sum_{i=1}^{h^\Delta} \lfloor n/2^{h^\Delta - i+1}/2 \rfloor + 1 = n/2$,

since, by construction, $d_i \geqslant d_{i-1}$. □

An example of Gabriel drawing of a ternary tree obtained by applying our algorithm is shown in Fig. 9. It is easy to see that for $\beta > 1$ the above construction does not guarantee the proximity regions of slanted edges to be empty. In fact, the third condition of quasi-Gabriel drawing definition does not prevent the lune of influence, for $\beta > 1$, of two adjacent vertices from being empty when layers are spaced out. However, as we will see in the next section, it is possible to modify the method described in Section 2 so to obtain β-proximity drawings of binary trees.

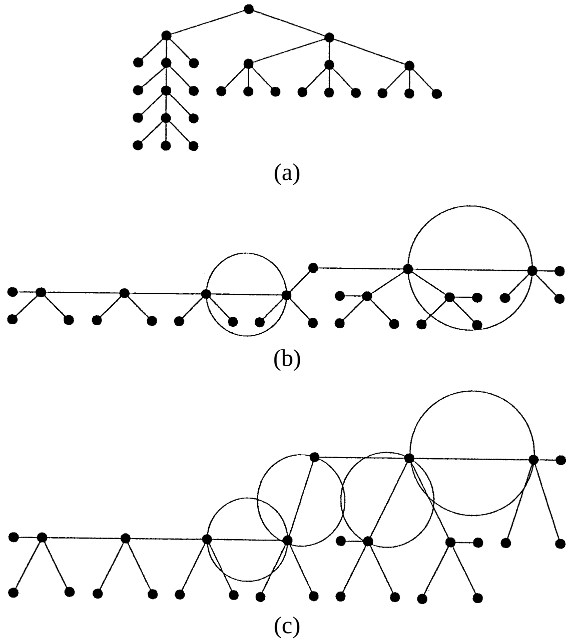

Fig. 9. Ternary trees: an example. (a) The tree given in input. (b) The quasi-Gabriel drawing Δ. (c) The weak Gabriel drawing.

3.2. β-proximity drawings of binary trees

In this section, we describe an algorithm to construct $O(n^2)$-area β-proximity drawings of binary trees. We make use of the technique described for ternary trees suitably modified. In particular we modify the definition of quasi-Gabriel drawing by imposing the edges to be represented with either horizontal or vertical segments. This gives rise to the definition of *quasi-proximity* drawing which allows us to consider β-proximity drawings for $0 \leqslant \beta < \infty$. We then present a linear-time algorithm to construct the quasi-proximity drawing of binary trees. As a consequence, given any binary tree with n nodes, we can construct polynomial-area weak β-proximity grid drawing in linear time.

Definition 3.4. A drawing Δ is a *quasi-proximity drawing* if it satisfies the following constraints:

(1) *Layered*. Vertices lie on layers.
(2) *No Transitive Edges*. Vertices on non-consecutive layers are not adjacent.
(3) *Orthogonal*. Edges are represented as horizontal or vertical segments.

Before presenting the extension of Theorem 2.4 we need a further definition. Let u, v and z be three points, we define:

$$\alpha(\beta) = \inf\{\angle uzv \mid z \in R[u, v, \beta]\} = \begin{cases} \arcsin \beta & \text{for } 0 \leqslant \beta < 1, \\ \arccos(1 - \frac{1}{\beta}) & \text{otherwise}. \end{cases}$$

In the proof we make use of the following quantity: $\delta(\beta) = 1/\tan(\alpha(\beta)/2)$. Intuitively, $\delta(\beta)$ represents the minimum distance such that for a unit-length horizontal edge (u, v), $R[u, v, \beta]$ does not intersect any layer L at distance $\delta(\beta)/2$ from L_u. Lemma 3.5 is a simple generalization of Lemma 2.3.

Lemma 3.5. *Let $\beta \geqslant 0$ and let u and v be any two vertices both laying on L_u and let L be a layer whose distance from L_u is bigger than $\delta(\beta) \cdot d(L_u)/2$. Then, it holds that $R[u, v, \beta] \cap L = \emptyset$.*

Similarly to Theorem 2.4, given a quasi-proximity drawing Δ, let h^Δ denote the number of layers and let d_i be the longest projection on layer l_i among edges whose at least one endpoint belongs to l_i. Notice that, for quasi-proximity drawings d_i is equal to the longest horizontal edge drawn on layer l_i.

We are now in a position to prove the following result, which is an extension of Theorem 2.4 to β-proximity drawings.

Theorem 3.6. *Let Δ be a quasi-proximity (grid) drawing. For any $0 \leqslant \beta < \infty$, a weak β-proximity (grid) drawing Δ_β exists such that:*

- $\text{width}(\Delta_\beta) = \text{width}(\Delta)$;
- $\text{depth}(\Delta_\beta) = \text{depth}(\Delta)$;
- $\text{height}(\Delta_\beta) \leqslant 2\delta(\beta) \sum_{i=1}^{h^\Delta} \lfloor d_i/2 \rfloor + 1$.

Moreover, if $d_i \geqslant d_{i-1}$ for $2 \leqslant i \leqslant h$, then $\text{height}(\Delta_\beta) \leqslant \delta(\beta) \sum_{i=1}^{h^\Delta} \lfloor d_i/2 \rfloor + 1$.

Proof. The proof is similar to that of Theorem 2.4. Let us denote by δ_i the distance between layer i and layer $i-1$ in Δ_β, for $2 \leqslant i \leqslant h^\Delta$. We define

$$\delta_i = \max\{\lfloor \delta(\beta)d_{i-1}/2\rfloor + 1, \lfloor \delta(\beta)d_i/2\rfloor + 1\}.$$

Thus

$$\text{height}(\Delta_\beta) \leqslant \sum_{i=2}^{h^\Delta} \delta_i < 2\delta(\beta)\sum_{i=1}^{h^\Delta} \lfloor d_i/2\rfloor + 1.$$

Moreover, if $d_i \geqslant d_{i-1}$ for $2 \leqslant i \leqslant h^\Delta$, then

$$\text{height}(\Delta_\beta) < \delta(\beta)\sum_{i=2}^{h^\Delta} \lfloor d_i/2\rfloor + 1.$$

In order to prove that Δ_β is a weak β-proximity drawing we show that the region of influence $R[u, v, \beta]$ of any two adjacent vertices does not contain any other vertex z. Let us first observe that, from Lemma 3.5, if a vertex z is contained in $R[u, v, \beta]$, then either $z \in L_u$ or $z \in L_v$.

Without loss of generality we assume $z \in L_u$ and we distinguish the following two cases:

- (u, v) *is a horizontal edge*. In this case $R[u, v, \beta] \cap L_u$ is the segment itself. This clearly implies that $z \notin R[u, v, \beta]$.
- (u, v) *is a vertical edge*. In this case, $R[u, v, \beta] \cap L_u = u$, which implies $z \notin R[u, v, \beta]$.

Hence the theorem follows. □

Motivated by the previous result we can now turn our attention to the construction of polynomial-area quasi-proximity drawings of binary trees.

Similarly to ternary trees, the construction of a quasi-proximity grid drawing Δ can be carried out recursively. In particular, we use the well-known recursive construction of so called h-v drawings [11, 19,40]. We denote with $\Delta_1 \ominus \Delta_2$ the drawing obtained by combining drawings Δ_1 and Δ_2 as follows: Δ_1 is translated to the bottom by one unit and Δ_2 is translated to the right by as many grid points as the width of Δ_1 plus 1 (see Fig. 10). It is easy to see that Δ is a quasi-proximity grid drawing and can be constructed in linear time. Moreover, its width is at most equal to the size n of the tree. An example of a quasi-proximity drawing is depicted in Fig. 11(b).

The following result can be proved similarly to Lemma 3.2.

Lemma 3.7. *For any* $2 \leqslant i \leqslant h^\Delta$, $d_i \leqslant \delta(\beta)n/2^{h^\Delta - i+1}$.

From Lemma 3.7 and Theorem 3.6 we obtain the following result.

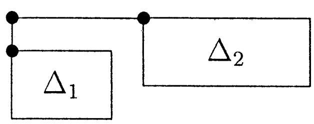

Fig. 10. The h-v drawing $\Delta_1 \ominus \Delta_2$ [11,19,40].

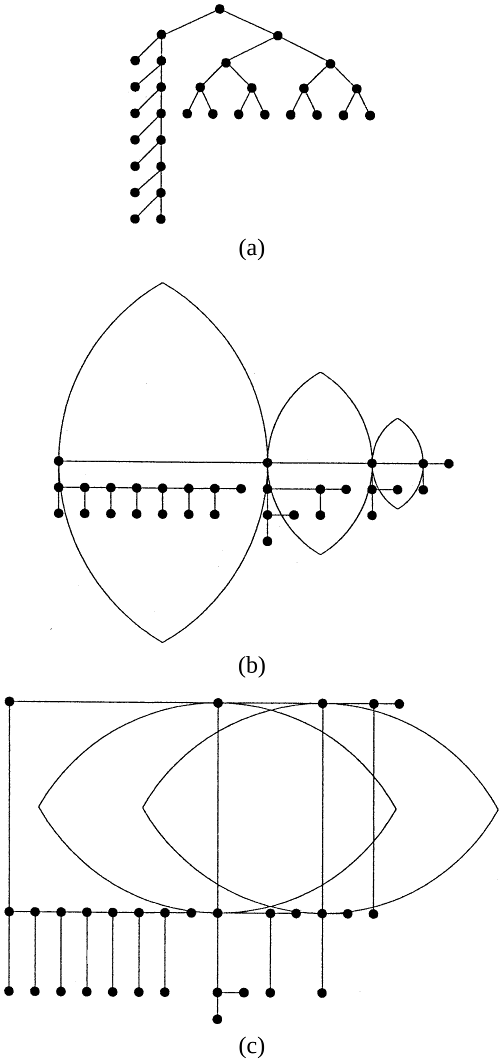

Fig. 11. Binary trees: an example. (a) The tree given in input. (b) The quasi-proximity drawing. (c) The β-proximity drawing with $\beta = 2$.

Corollary 3.8. *For any $0 \leqslant \beta < \infty$ and for any binary tree T with n nodes, a weak β-proximity grid drawing of $O(\delta(\beta)n^2)$-area exists, which can be constructed in $O(n)$ time.*

4. Proximity drawings in 3D-space

This section is devoted to the construction of proximity drawings in the 3-dimensional space. As we will prove in the sequel, the use of the third dimension, combined with the method described in Section 2, allows to design efficient proximity drawing algorithms. Indeed, we will prove that it is possible to construct 3-dimensional weak Gabriel drawings of unbounded degree trees in n^4 volume. Notice that unbounded degree trees are not strong drawable [29]. Moreover, we will show a class of graphs, requiring exponential area for weak Gabriel drawings, that admits linear-volume strong β-proximity drawing instead, for any $1 \leqslant \beta \leqslant 2$.

4.1. Unbounded degree trees

In this section we consider unbounded degree trees and prove that they admit n^4-volume weak Gabriel drawings. To this aim we will show how to construct a quasi-Gabriel drawing Δ whose volume is n^3 and such that any edge has length at most $n/\sqrt{2}$.

We denote by x_u, y_u and z_u, the x-, y- and z-coordinates of a vertex u. The construction of Δ takes two steps.

4.1.1. Step 1: front drawing

In the first phase we construct an upward straight-line layered drawing of T on the yz-plane (i.e., all the vertices have null x coordinate).

We want our drawing to satisfy the following invariant: *Each internal vertex is at the same distance from its leftmost and its rightmost child.*

Let T be a tree having as immediate subtrees $T_1, \ldots, T_k$. The algorithm in Fig. 13 correctly computes the front drawing of T in linear time (see also Fig. 12(a)) which satisfies the above stated invariant.

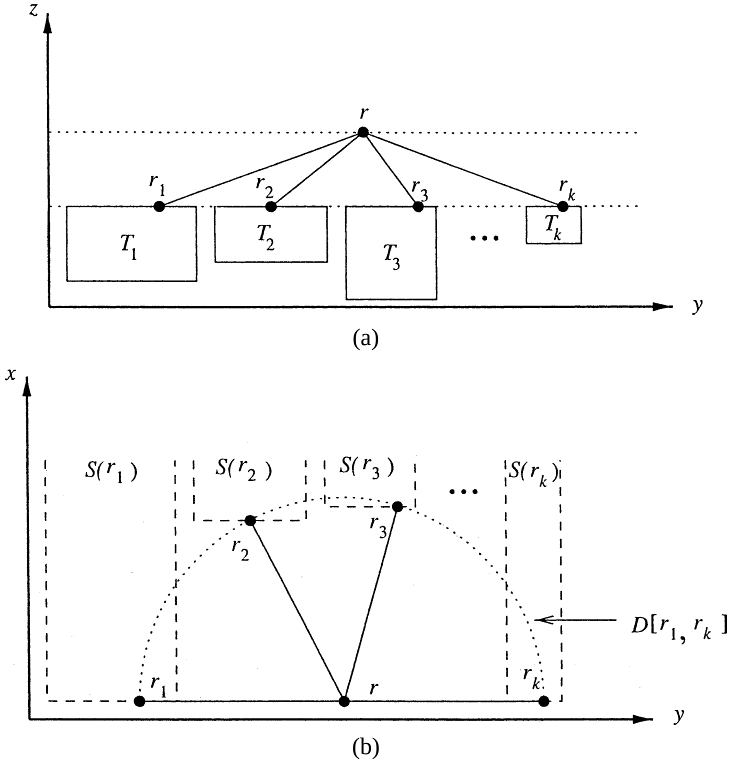

Fig. 12. The two steps of the construction of 3-dimensional proximity drawings of unbounded degree trees: (a) the front drawing; (b) equally space the children of each node.

```
algorithm front_drawing(T)
h ← height of T
r ← root of T
if h = 1 then
      draw r on layer 1
else begin
      T_1 ← largest immediate subtree of T
      r_1, ..., r_k ← roots of T_1, ..., T_k children of r
      for i = 1 to k do
            Δ_i = front_drawing(T_i)
      translate Δ_1 so that r_1 is on layer h − 1
      for i = 2 to k do
            translate Δ_i so that:
                  1. r_i is on layer h − 1, and
                  2. Δ_i is at unit horizontal distance from Δ_{i−1}
      draw r on layer h at the same distance from r_1 and r_k
      connect r to r_1, ..., r_k
      end
end
```

Fig. 13. Step 1: Algorithm `front-drawing`.

```
algorithm move(T)
h ← height of T
r ← root of T
if h = 1 then
      draw r on layer 1
r_1, ..., r_k ← roots of T_1, ..., T_k children of r
d = d(r_1, r_k)
for i = 2 to k − 1 do begin
      x_{r_i} = x_r + sqrt(d^2/2 − (y_r − y_{r_i})^2)
end
for i = 1 to k do begin
      move(T_{r_i})
end
```

Fig. 14. Step 2: Algorithm `move`.

4.1.2. Step 2: equally space the children

Let v be an internal vertex of T and $v_1, \ldots, v_k$ be its children. In this step we assign different x-coordinates to vertices $v_1, \ldots, v_k$ so that all edges (v, v_i) satisfy the third condition of the quasi-Gabriel definition. In particular, we assign different x-coordinates to vertices $v_1, \ldots, v_k$ so that all edges (u, v_i), with $1 \leqslant i \leqslant k$, have the same length. Let $D[v_1, v_k]$ be the disk on the layer L_{v_1} containing $v_1, \ldots, v_k$ and having as antipodal points v_1 and v_k (see Fig. 12(b)). We translate $v_2, \ldots, v_{k-1}$ along the x-direction until they meet the boundary of $D[v_1, v_k]$ (see Fig. 12(b)). Algorithm `move`(T) in Fig. 14 implements the above strategy in linear time.

4.1.3. Proof of correctness

For any tree T given in input, let us denote with Δ the drawing of T obtained according to the two steps previously described. We first prove that the volume of Δ is polynomial.

Lemma 4.1. *For any n nodes tree T, the drawing Δ has volume at most n^3.*

Proof. It is easy to see that the height and the width of Δ are at most n. Let us consider the depth of Δ and prove by induction on n that it is at most n.

Step base $(n = 1)$. Trivial.

Inductive step. Let us suppose that the lemma holds for all trees with at most $n - 1$ nodes, and let T be an n node tree. Let $T_1, \ldots, T_k$ be its immediate subtrees, and let $n_1, \ldots, n_k$ be their size, respectively, with $n_1 \geqslant n_2 \geqslant \cdots \geqslant n_k$. We denote by $\Delta_1, \ldots, \Delta_k$ the drawings of $T_1, \ldots, T_k$, respectively. Δ is obtained by combining these subdrawings as shown in Fig. 12(b), where $r_1, \ldots, r_k$ denote the roots of $T_1, \ldots, T_k$, respectively. Since $d(r_1, r_k)$ is at most n, each r_i, for $2 \leqslant i \leqslant k - 1$, is translated along the x-direction by at most $n/2$. Thus, by inductive hypothesis and considering that $n_i \leqslant n/2$, for $2 \leqslant i \leqslant k$, we obtain

$$\mathrm{depth}(\Delta) \leqslant \max\bigl\{\mathrm{depth}(\Delta_1), n/2 + \mathrm{depth}(\Delta_2), \ldots, n/2 + \mathrm{depth}(\Delta_k)\bigr\} \leqslant \max\{n_1, n\} = n.$$

The lemma thus follows. □

In order to prove that the drawing Δ is a quasi-Gabriel drawing we make use of the following intermediate result.

Lemma 4.2. *Let $S(u)$ denote the smallest isothetic parallelepiped containing the drawing of the subtree rooted at u. Also let u and v and z be any three vertices such that*: (a) *u is a child of v*; *and* (b) *the subtrees rooted at v and z are disjoint. Then, it holds that $R_u[u, v] \cap S(z) = \emptyset$.*

Proof. Let $v_1, v_2, \ldots, v_k$ denote the children of v. By construction, $R_{v_i}[v, v_i]$ is contained in $D[v_1, v_k]$ (see Fig. 12(b)). Moreover, $D[v_1, v_k]$ is contained in the strip determined by the smallest and largest x-coordinate of $S(v)$. The recursive construction performed by algorithms `front_drawing` and `move` easily implies that $S(v) \cap S(z) = \emptyset$. Hence the lemma follows. □

Lemma 4.3. *For any n nodes tree T, the drawing Δ is a quasi-Gabriel drawing.*

Proof. It is easy to see that Δ is a layered drawing with no transitive edges. Thus, it remains to prove that for any edge (u, v), $R_u[u, v] \cap L_u$ contains no vertices other than u and v. The proof is by induction on the number n of nodes of the tree.

Base step $(n = 1)$. Trivial.

Inductive step. Let us assume that the theorem holds for any tree with at most $n - 1$ nodes, and let us consider an n nodes tree T. We distinguish the following two subcases:

- $v = r$. In this case $u = r_i$, for some $1 \leqslant i \leqslant k$. Also, layer L_u contains the children of r only. Let r' be the projection of r on L_u. By construction, $d(r', r_i) = d(r', r_j)$, thus implying that $\angle r' r_i r_j < \pi/2$, for any $i \neq j$. Hence, $r_j \notin R_u[u, v]$, for any $j \neq i$.
- $u = r$. In this case, we simply observe that u is the only vertex L_u.

- $u, v \neq r$. Let z be any vertex other than u in L_u. If v is a child of u, we can apply Lemma 4.2 and obtain $z \notin R_u[u, v]$. Otherwise, that is u is a child of v, Lemma 4.2 implies that $z \in R_u[u, v]$ only if z is a child of v as well. In the latter case, the same proof as the case $v = r$ above applies. □

By combining Theorem 2.4, Lemmas 4.1 and 4.3 and considering that the length of any edge is bounded by $n/\sqrt{2}$, we obtain the following result.

Theorem 4.4. *For any tree T with n nodes there exists a weak Gabriel drawing whose volume is at most n^4 with $O(\log n)$ bit-requirement. Moreover, the drawing can be constructed in linear time.*

4.2. Exponential area versus polynomial volume

In this section we consider an infinite class of graphs introduced in [33]. In [30] the authors proved an exponential-area lower bound for β-proximity drawings, for $1 \leqslant \beta \leqslant 1/(1 - \cos 2\pi/5) \simeq 1.45$.

We apply the method described in Section 2 and we show that this class admits a linear volume strong β-proximity drawing, for any $1 \leqslant \beta < 2$, and a linear volume relative neighborhood drawing.

4.2.1. Class of graphs

The class is recursively defined as follows. Graph G_1 is the graph shown in Fig. 15(a). The graph G_{i+1} is obtained from G_i by adding five vertices $v_1^{i+1} v_2^{i+1} v_3^{i+1} v_4^{i+1} v_5^{i+1}$ and by connecting them to G_i as shown in Fig. 15(b). Clearly, the number of nodes of G_n is $5n + 1$. We denote with P_i the pentagon of G_i given by the 5-cycle $v_1^i v_2^i v_3^i v_4^i v_5^i$. Notice that each side of pentagon P_i forms a triangle with a vertex of P_{i+1}, as well as each side of P_{i+1} with a vertex in P_i. We refer to these triangles as *petals*.

Theorem 4.5 [30]. *A Gabriel drawing and a weak Gabriel drawing of graph G_n require area $\Omega(3^n)$, under any resolution rule assumption.*

In the same paper, the authors generalized the previous result to β-drawings, for any $1 \leqslant \beta < 1/(1 - \cos 2\pi/5)$.

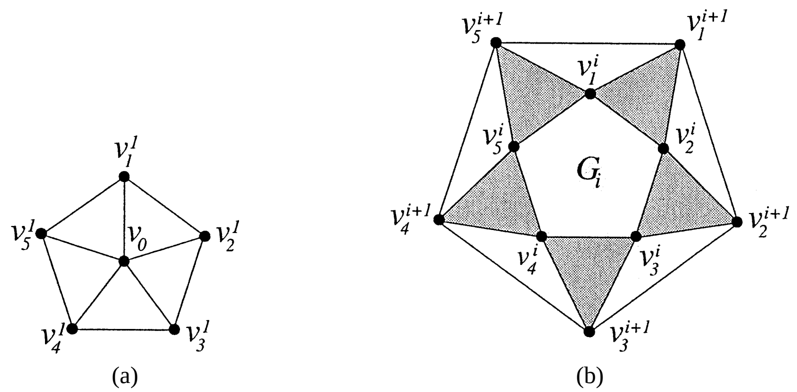

Fig. 15. The exponential-area/linear volume class: (a) graph G_1; (b) graph G_{i+1} given G_i.

4.2.2. Construction of the drawings

In this section we describe a linear-time algorithm to construct a *linear-volume strong Gabriel* drawing of G_n.

To this aim we will first describe how to construct a linear-volume quasi-Gabriel drawing of G_n such that the maximum length of any edge is constant. This implies that by suitably choosing a constant distance δ between consecutive layers G_n admits a linear-volume weak Gabriel drawing. In the next section we will prove the correctness of the algorithm and we will show how to extend it to strong proximity.

The construction of the drawing is defined as follows: Pentagon P_i, for $1 \leqslant i \leqslant n$, is drawn on layer i as a regular pentagon. Moreover, P_{i+1} is rotated by a $\pi/5$ angle with respect to P_i (see Figs. 16(a) and 16(c) which show a drawing of G_4). Notice that since the distance between consecutive layers is constant and each pentagon P_i is drawn in constant area, the volume is $O(n)$. It is easy to see that the algorithm `pentagons` described in Fig. 17 implements the above strategy in linear time.

4.2.3. Proof of correctness

In order to prove the correctness of the algorithm we first show that the resulting drawing is a quasi-Gabriel drawing. This implies that, by suitably choosing the constant δ, it can be transformed into a linear-volume weak Gabriel drawing.

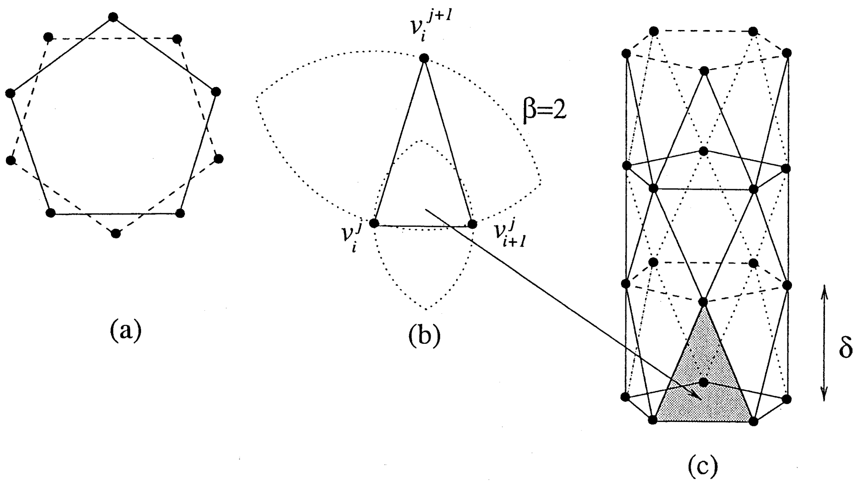

Fig. 16. (a) Two consecutive pentagons viewed from the top. (b) How to draw a single petal. (c) The whole 3-dimensional drawing.

algorithm `pentagons`(G_n)
draw G_1 on layer 1 such that P_1 is a regular pentagon centered at v_0
for $i = 2$ **to** n **do begin**
  draw P_i on layer i rotated by $\pi/5$ with respect to P_{i-1}
  connect P_i with P_{i-1}
end

Fig. 17. The algorithm to draw G_n in linear volume.

Lemma 4.6. *For any n the algorithm* `pentagons` *returns an* $\mathrm{O}(n)$*-volume quasi-Gabriel drawing.*

Proof. Let us first observe that the drawing satisfies the first two properties of Definition 2.1 of quasi-Gabriel drawing. Thus we have to prove that for any edge (u, v), $R_u[u, v]$ contains no vertices except for u and v. By construction, the following two cases arise:

(1) *u and v are on the same layer.* Let P_i be the pentagon containing u and v. Since P_i is drawn as a regular pentagon (see Fig. 16(a)), then, obviously, the theorem holds.
(2) *u and v are on consecutive layers.* Without loss of generality let $u \in P_i$ and $v \in P_{i+1}$. Again, since P_i and P_{i-1} are drawn as regular polygons and P_{i-1} is rotated by $\pi/5$ (see Fig. 16(a)), it is easy to see that $R_u[u, v]$ does not contain any vertex of P_i other than u. □

Let us observe that, since pentagons are equally drawn on consecutive layers at unitary distance, then the maximum edge length is constant. By Theorem 2.4 we obtain the following result.

Theorem 4.7. *For any n, graph G_n admits an* $\mathrm{O}(n)$*-volume weak Gabriel drawing.*

Proof. The proof follows by Theorem 2.4 choosing the distance δ between layers equal to d_i, where d_i is the length of an edge of P_i. □

The above result can be extended to strong Gabriel drawings.

Theorem 4.8. *For any n, graph G_n admits a* $\mathrm{O}(n)$*-volume strong Gabriel drawing.*

Proof. Let Δ be the weak Gabriel drawing obtained by algorithm `pentagons` where layers are spaced out by the amount δ specified in Theorem 4.7. Let us first observe that, for any $\delta' \geqslant \delta$, the resulting drawing still is a weak Gabriel drawing. In the following, we will show that a constant $\delta' \geqslant \delta$ exists such that Δ is a strong Gabriel drawing for G_n. To this aim we have to prove that the proximity region $R[u, v]$ of any two non adjacent vertices u and v contains at least another vertex. We distinguish the following three cases:

(1) *u and v are not on consecutive layers.* Without loss of generality, we assume that u and v belong to P_i and P_{i+2}, respectively. It is then easy to see that for a sufficiently (but still constant) large δ' at least one vertex of P_{i+1} falls within $R[u, v]$. Notice that the value of δ' depends on the length of the side of the pentagon only.
(2) *u and v are on consecutive layers.* In this case consider the disk $D[u, v']$, where v' is the projection of v on layer L_u. Clearly, $D[u, v']$ contains at least another vertex of L_u (see Fig. 16(a)). This implies that $R[u, v]$ also contains the same vertex.
(3) *u and v are on the same layer.* By construction, the drawing of each P_i is a strong Gabriel drawing. □

In the following we show that the construction is even more powerful since it allows to derive strong β-proximity drawings, for $1 \leqslant \beta < 2$.

Theorem 4.9. *For any n, graph G_n admits a linear volume relative neighborhood drawing and strong β-proximity drawing, for any $1 \leqslant \beta < 2$.*

Proof. We first consider strong β-proximity drawings. We modify the drawing of G_1 since it is not β-drawable on a plane for $\beta \geqslant 1/(1 - \cos 2\pi/5)$ (see Fig. 15(a)). Translate v_0 on layer 0 so that it is at the same distance from all the vertices of P_1 (i.e., the new drawing of v_0 corresponds to the orthogonal projection on layer 0 of the old drawing of v_0). All other pentagons are drawn as described in Theorem 4.8.

Observe that for any $1 \leqslant \beta < 2$ and for any two vertices u and v

$$R[u, v] \subseteq R[u, v, \beta].$$

This implies that for any $\delta_\beta \geqslant \delta'$, where δ' is the value defined in the proof of Theorem 4.8, the construction yields a drawing such that the proximity region of any two non-adjacent vertices contains at least another vertex. Thus, in order to prove the theorem it suffices to show that for any two adjacent vertices u and v, the proximity region $R[u, v, \beta]$ is empty. Let d be the maximum edge length in the drawing of Theorem 4.8 and let $\delta_\beta = d\delta(\beta)$, where $\delta(\beta)$ is as defined in Section 3.2. (Notice that the value of δ_β is proportional to l and does not depend on n.) We consider the following two cases:

(1) *u and v belongs to the same pentagon.* Using basic geometry it is possible to prove that no other vertex of the pentagon containing u and v can fall within $R[u, v, \beta]$. Additionally, because of the choice of δ_β and by Lemma 3.5, no vertex from other pentagons falls within $R[u, v, \beta]$.
(2) *u and v belongs to consecutive pentagons.* Let P_i and P_{i+1} be the two pentagons containing u and v, respectively. Again, because of the choice of δ_β and by Lemma 3.5, any vertex not belonging to P_i and to P_{i+1} cannot be contained on $R[u, v, \beta]$. Without loss of generality, let us consider a vertex z of P_i. If z is adjacent to u and v then u, v and z form a petal (see Figs. 15 and 16(b)). By considering the plane containing the three vertices and its intersection with $R[u, v, \beta]$ (see Fig. 16(b)) it is easy to see that z lays on the boundary of $R[u, v, 2]$. Hence, for any $\beta < 2$, $z \notin R[u, v, \beta]$. Similarly, we can prove that no other vertex of P_i is contained in $R[u, v, \beta]$. The same holds for z in P_{i+1}.

Finally, the above considerations also apply to relative neighborhood drawings. Indeed, a relative neighborhood drawing is a slight modification of strong 2-proximity drawings, where the proximity region is defined as the intersection of two *open spheres* [31]. The theorem thus follows. □

5. Conclusions and open problems

In this paper we have introduced a novel technique to construct proximity drawings. By applying our technique to trees, we obtain the first algorithms that construct drawings whose size is polynomial in the number of the vertices. We also gave some evidence that our method is quite powerful, since it allows to construct linear volume proximity drawings of a class of graphs that requires exponential area, instead.

Several problems are left open by this paper. They mainly concern the construction of polynomial size proximity drawings and the study of other classes of graphs to which apply our method. In particular, the following research directions seem to us the more promising:

- *Extend the results to other classes of graphs.* As for as the 2-dimensional case, it might be interesting to consider weak Gabriel drawings of trees of degree 4. It is worth observing that even ternary trees do not admit *strong Gabriel* drawings [4]. Moreover, do ternary trees admit β-proximity drawings of polynomial area for some $\beta > 1$? As for the 3-dimensional case it could be interesting to consider other classes of β-drawable graphs in the plane such as outerplanar graphs.
- *Consider strong proximity.* Do binary trees admit *at least* 3-*dimensional* strong Gabriel drawings of polynomial volume?
- *Prove lower bounds.* A related issue is that of proving a lower bound on the area of trees for both weak and strong proximity. In particular, are the algorithms given in Section 3 optimal?

References

[1] H. Alt, M. Godau, S. Whitesides, Universal 3-dimensional visibility representations for graphs, in: F.J. Brandenburg (Ed.), Graph Drawing (Proc. GD '95), Lecture Notes in Computer Science, vol. 1027, Springer, Berlin, 1996, pp. 8–19.

[2] T.C. Biedl, Heuristics for 3d-orthogonal graph drawings, in: 4th Twente Workshop on Graphs and Combinatorial Optimization, 1995, pp. 41–44.

[3] P. Bose, G. Di Battista, W. Lenhart, G. Liotta, Proximity constraints and representable trees, in: Graph Drawing (Proc. GD '94), Lecture Notes in Computer Science, vol. 894, Springer, Berlin, 1994, pp. 340–351.

[4] P. Bose, W. Lenhart, G. Liotta, Characterizing proximity trees, Algorithmica 16 (1996) 83–110.

[5] M. Brown, M. Najork, Algorithm animation using 3d interactive graphics, in: Proc. ACM Symp. on User Interface Software and Technology, 1993, pp. 93–100.

[6] I. Cahit, Drawing the complete graph in 3-D with straight lines and without crossings, Bull. ICA (1994) 12.

[7] T. Calamoneri, A. Sterbini, Drawing 2-, 3- and 4-colorable graphs in $O(n^2)$ volume, in: S. North (Ed.), Graph Drawing (Proc. GD '96), Lecture Notes in Computer Science, vol. 1190, Springer, Berlin, 1997, pp. 53–62.

[8] T. Chang, M.T. Goodrich, S.R. Kosaraju, R. Tamassia, Optimizing area and aspect ratio in straight-line orthogonal tree drawings, in: Graph Drawing (Proc. GD '96), Lecture Notes in Computer Science, vol. 1190, 1996, pp. 63–75.

[9] M. Chrobak, M.T. Goodrich, R. Tamassia, Convex drawings of graphs in two and three dimensions, in: Proc. 12th Annual ACM Sympos. Comput. Geom., 1996, pp. 319–328.

[10] R.F. Cohen, P. Eades, T. Lin, F. Ruskey, Three-dimensional graph drawing, in: R. Tamassia, I.G. Tollis (Eds.), Graph Drawing (Proc. GD '94), Lecture Notes in Computer Science, vol. 894, Springer, Berlin, 1995, pp. 1–11.

[11] P. Crescenzi, G. Di Battista, A. Piperno, A note on optimal area algorithms for upward drawings of binary trees, Computational Geometry 4 (5) (1992) 235–282.

[12] P. Crescenzi, P. Penna, Strictly-upward drawings of ordered search trees, Theoret. Comput. Sci. 203 (1998) 51–67.

[13] P. Crescenzi, P. Penna, A. Piperno, Linear area upward drawings of AVL trees, Computational Geometry 9 (1998) 25–42.

[14] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis, Algorithms for drawing graphs: an annotated bibliography, Computational Geometry 4 (1994) 235–282.

[15] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis, Graph Drawing—Algorithms for Drawing Graphs, Prentice Hall, Englewood Cliffs, NJ, 1999.

[16] G. Di Battista, W. Lenhart, G. Liotta, Proximity drawability: a survey, in: Graph Drawing (Proc. GD '94), Lecture Notes in Computer Science, vol. 894, Springer, Berlin, 1994, pp. 328–339.

[17] G. Di Battista, G. Liotta, S.H. Whitesides, The strength of weak proximity, in: F.J. Brandenburg (Ed.), Graph Drawing (Proc. GD '95), Lecture Notes in Computer Science, vol. 1027, Springer, Berlin, 1996, pp. 178–189.

[18] G. Di Battista, R. Tamassia, I.G. Tollis, Area requirements and symmetry display of planar upward drawings, Discrete Comput. Geom. 7 (1992) 381–401.

[19] P. Eades, T. Lin, X. Lin, Minimum size h-v drawings, in: Proc. of Internat. Workshop AVI'92, 1992, pp. 386–395.

[20] P. Eades, T. Lin, X. Lin, Two tree drawing conventions, Internat. J. Comput. Geom. Appl. 3 (1996) 133–153.

[21] P. Eades, S. Whitesides, The realization problem for Euclidean minimum spanning tree is NP-hard, Algorithmica 16 (1996).

[22] H. ElGindy, G. Liotta, A. Lubiw, H. Meijer, S.H. Whitesides, Recognizing rectangle of influence drawable graphs, in: Graph Drawing (Proc. GD ’94), Lecture Notes in Computer Science, vol. 394, Springer, Berlin, 1994, pp. 352–363.
[23] K.R. Gabriel, R.R. Sokal, A new statistical approach to geographic variation analysis, Systematic Zoology 18 (1969) 259–278.
[24] A. Garg, M.T. Goodrich, R. Tamassia, Area-efficient upward tree drawing, in: Proc. ACM Symp. on Computational Geometry, 1993, pp. 359–368.
[25] A. Garg, R. Tamassia, On the computational complexity of upward and rectilinear planarity testing, in: Graph Drawing (Proc. GD ’94), Lecture Notes in Computer Science, vol. 894, 1994, pp. 286–297.
[26] A. Garg, R. Tamassia, GIOTT03D: A system for visualizing hierarchical structures in 3d, in: S. North (Ed.), Graph Drawing (Proc. GD ’96), Lecture Notes in Computer Science, vol. 1190, 1997, pp. 193–200.
[27] D.G. Kirkpatrick, J.D. Radke, A framework for computational morphology, in: G.T. Toussaint (Ed.), Computational Geometry, North-Holland, Amsterdam, 1985, pp. 217–248.
[28] W. Lenhart, G. Liotta, Proximity drawings of outerplanar graphs, in: S. North (Ed.), Graph Drawing (Proc. GD ’96), Lecture Notes in Computer Science, vol. 1190, 1997, pp. 286–302.
[29] G. Liotta, G. Di Battista, Computing proximity drawings of trees in the 3-dimensional space, in: Proc. 4th Workshop Algorithms Data Struct., Lecture Notes in Computer Science, vol. 955, Springer, Berlin, 1995, pp. 239–250.
[30] G. Liotta, R. Tamassia, I.G. Tollis, P. Vocca, Area requirement of Gabriel drawings, in: G. Bongiovanni, D.P. Bovet, G. Di Battista (Eds.), Proc. CIAC’97, Lecture Notes in Computer Science, vol. 1203, Springer, Berlin, 1997, pp. 135–146.
[31] A. Lubiw, N. Sleumer, All maximal outerplanar graphs are relative neighborhood graphs, in: Proc. CCCG’93, 1993, pp. 198–203.
[32] J. MacKinley, G. Robertson, S. Card, Cone trees: Animated 3d visualizations of hierarchical information, in: Proc. of SIGCHI Conf. on Human Factors in Computing, 1991, pp. 189–194.
[33] D.W. Matula, R.R. Sokal, Properties of Gabriel graphs relevant to geographic variation research and clustering of points in the plane, Geogr. Anal. 12 (1980) 205–222.
[34] P.T. Metaxas, G.E. Pantziou, A. Symvonis, A note on parallel algorithms for optimal h-v drawings of binary trees, Computational Geometry 9 (1998) 145–158.
[35] C. Monma, S. Suri, Transitions in geometric minimum spanning trees, Discrete Comput. Geom. 8 (1992) 265–293.
[36] M. Patrignani, F. Vargiu, 3DCube: A tool for three dimensional graph drawing, in: G. Di Battista (Ed.), Graph Drawing (Proc. GD ’97), Lecture Notes in Computer Science, vol. 1353, Springer, Berlin, 1997, pp. 284–290.
[37] P. Penna, On the approximability of two tree drawing conventions, Inform. Process. Lett. 82 (5) (2002) 237–242.
[38] J.D. Radke, On the shape of a set of points, in: G.T. Toussaint (Ed.), Computational Morphology, North-Holland, Amsterdam, 1988, pp. 105–136.
[39] S. Reiss, An engine for the 3D visualization of program information, J. Visual Languages and Computing 6 (3) (1995).
[40] S. Shiloach, Linear and planar arrangements of graphs, PhD Thesis, Department of Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel, 1976.
[41] G.T. Toussaint, The relative neighborhood graph of a finite planar set, Pattern Recogn. 12 (1980) 261–268.
[42] R.B. Urquhart, Graph theoretical clustering based on limited neighborhood sets, Pattern Recogn. 15 (1982) 173–187.
[43] L. Valiant, Universality considerations in VLSI circuits, IEEE Trans. Comput. C-30 (2) (1981) 135–140.