Proximity Drawings: Three Dimensions Are Better than Two*
(Extended Abstract)

Paolo Penna[1] and Paola Vocca[2]

[1] Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza", Via Salaria 113, I-00198 Rome, Italy.
penna@mat.uniroma2.it,

[2] Dipartimento di Matematica, Università di Roma "Tor Vergata", Via della Ricerca Scientifica, I-00133 Rome, Italy.
vocca@axp.mat.uniroma2.it

Abstract. We consider weak Gabriel drawings of unbounded degree trees in the three-dimensional space. We assume a minimum distance between any two vertices. Under the same assumption, there exists an exponential area lower bound for general graphs. Moreover, all previously known algorithms to construct (weak) proximity drawings of trees, generally produce exponential area layouts, even when we restrict ourselves to binary trees. In this paper we describe a *linear-time polynomial-volume* algorithm that constructs a *strictly-upward weak Gabriel drawing* of any *rooted tree* with $O(\log n)$-bit requirement. As a special case we describe a Gabriel drawing algorithm for binary trees which produces integer coordinates and n^3-*area* representations . Finally, we show that an infinite class of graphs requiring exponential area, admits *linear-volume Gabriel drawings*. The latter result can also be extended to β-drawings, for any $1 < \beta < 2$, and relative neighborhood drawings.

1 Introduction.

Three–dimensional drawings of graphs have received increasing attention recently due to the availability of low–cost workstations and of applications that require three–dimensional representations of graphs [6, 13, 18, 22, 20]. Even though there are several theoretical results [1, 7, 8, 9], there is still the need for a better theoretical understanding of three-dimensional capabilities.

In this paper we tackle the problem of drawing proximity drawings in the three–dimensional space. Proximity drawings have been deeply investigated in the two–dimensional space because of their interesting graphical features (see, e.g. [17, 2, 5, 10, 11, 4, 16]). Nevertheless, only preliminary results are available for three–dimensional proximity drawings [15].

* Work partially supported by the Italian Project MURST-"Algorithms and Data Structure"

S.H. Whitesides (Ed.): GD'98, LNCS 1547, pp. 275–287, 1998.

In the three–dimensional space, a *proximity drawing* is a straight–line drawing where two vertices are adjacent if and only if they are *neighbors* according to some definition of *neighborhood*. One way of defining a neighborhood constraint between a pair of vertices is to use a *proximity region*, that is a suitable region of the plane having the two points on the boundary. 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 drawing (however, an edge of the drawing may cross the proximity region). For example, two vertices u and v are considered to be neighbors if and only if the closed disk having u and v as antipodal points, is empty. Proximity drawings that obey this neighborhood constraint are known in the literature as *Gabriel drawings* ([12, 19]) and the closed disk is called *Gabriel disk*. In a *relative neighborhood drawing* [23] two vertices u and v are adjacent if there is no other point whose distance is strictly less than the Euclidean distance between u and v. A generalization of Gabriel and relative neighborhood drawings is represented by *β–drawings*, where the proximity region is defined by the parameter β. β–drawings were first introduced by Kirkpatrick and Radke [14, 21] in the computational morphology context.

In the three–dimensional space the definition of proximity drawings is an obvious extension of that for the two dimensional case. In particular, we refer to Gabriel proximity region as a *Gabriel sphere* instead of disk.

In [3], the *weak proximity drawings* have been first introduced. 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.

As we will show in this paper, this weaker definition of proximity yields more efficient drawing algorithms while preserving the graphical features of strong proximity drawings (e.g. edges are represented as straight lines; vertices not incident to a certain edge are drawn far apart from that edge).

Unfortunately, all known algorithms that compute (weak) proximity drawings produce representations whose area increases exponentially with the number of vertices. As a consequence, the problem of constructing proximity drawings of graphs that have small area is considered a very challenging one by several authors (see [5, 10, 19]). Additionally, in [16] an exponential lower bound on the area of Gabriel drawings has been presented.

In this paper we shown that the use of the third dimension can substantially help in improving the efficiency of the proximity drawings. More precisely, the results we achieve in this paper are listed below:

– We describe a linear time polynomial volume algorithm for strictly upward weak Gabriel drawing of unbounded degree trees, where the coordinates of vertices can be represented with $O(\log n)$-bits;
– We give a n^3-area algorithm for strictly upward weak Gabriel drawing of binary trees where all vertices have integer coordinates;
– We present an infinite class of graphs such that any Gabriel drawing (both strong and weak) of a graph in the class requires area exponential in the number of vertices, while admits linear volume strong Gabriel drawing;

– We extend the above result to β-drawings for $1 \leq \beta < 2$ and to relative neighborhood drawings;

In all algorithms we present we assume a minimum distance between vertices, which imposes that dimensions cannot be arbitrarily scaled down.

2 Preliminaries.

A *three-dimensional layered drawing* of a tree T is a drawing such that each vertex is placed on equally spaced layers, being a *layer* a plane orthogonal to the z-axis. In the following we denote with δ the distance of any two consecutive layers and with layer i the plane given by the points whose z coordinate is equal to δi. Thus, a layered drawing is a drawing such that the z-coordinate of the vertices takes value in $\{\delta, 2\delta, \ldots, i\delta, \ldots\}$. The *height*, the *width* and the *depth* of a layered drawing are defined as the height, width and depth of the smallest isothetic parallelepiped bounding the drawing. Given a vertex a we denote with L_a the layer on which the vertex is drawn.

Given two points in the three-dimensional space, we denote with $R[a, b, \beta]$ the *β-region of influence of a and b*. For $0 < \beta < 1$, $R[a, b, \beta]$ is the intersection of the two closed spheres of radius $d(a, b)/(2\beta)$ passing through both a and b. For $1 \leq \beta < \infty$, $R[a, b, \beta]$ is the intersection of the two closed spheres of radius $\beta d(a, b)$ and centered at the points $(1-\beta/2)a + (\beta/2)b$ and $(\beta/2)a + (1-\beta/2)b$. A *weak β-drawing* for a graph G is a drawing of G such that for each pair of adjacent vertices a and b, the proximity region $R[a, b, \beta]$ does not contain any other vertex of the drawing. If the proximity region of any two *non adjacent* vertices contains at least another vertex of the drawing then the drawing of G is a *strong β-drawing* or simply *β-drawing.* A (weak) Gabriel drawing is a (weak) β-drawing with $\beta = 1$. In this case, the proximity region of any two points a and b is denoted by $R[a, b]$ and corresponds to the closed sphere centered at the middle point between a and b whose radius is $d(a, b)$. Proximity regions of a 2-dimensional drawing are similarly defined as the intersection of closed disks.

Finally, to simplify the notation, we denote a vertex and a point representing it with the same symbol. Additionally, let u be a vertex. We denote by x_u, y_u and z_u its x-, y-, and z-coordinates.

3 The Algorithm.

In this section we describe a linear-time n^4-volume algorithm that constructs a *strictly-upward weak Gabriel drawing* of any rooted tree T with n nodes. The correctness of the algorithm will be proved in Sect. 4.

The construction of the drawing is performed in two phases. In Phase 1 we construct an *upward straight-line layered drawing* of T in the yz plane. This will be the "front" of a three-dimensional drawing. Indeed, in Phase 2, we assign different x-coordinates to the vertices, so that the children of a vertex are at the same distance from the parent. This will be performed by simply moving the subdrawings along to the x-direction.

3.1 Phase 1: The Front Drawing.

In the first step we construct an upward straight–line layered drawing of T on the yz-plane (i.e. all the vertices have null x coordinate). The value of the distance δ of two consecutive layers will be specified in Sect. 4.

We want our drawing to satisfy the following two invariants:

1. Each edge connects vertices on consecutive layers.
2. Each internal vertex is at the same distance from its leftmost and its rightmost child.

---

algorithm `front_drawing`(T)
$h \leftarrow$ height of T
$r \leftarrow$ root of T
if $h = 1$ **then**
    draw r on layer 1 with null y-coordinate
else begin
    $T_1 \leftarrow$ largest immediate subtree of T
    $r_1, \ldots, r_k \leftarrow$ roots of $T_1, \ldots, T_k$ children of r
    for $i = 1$ **to** k **do**
        $\Delta_i =$`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 distance from Δ_{i-1}
    draw r on layer h at the same distance from r_1 and r_k
    connect r to $r_1, \ldots, r_k$
    end
end

Fig. 1. Phase 1: Algorithm `front_drawing`.

The algorithm in Fig. 1 constructs a drawing of tree T having as immediate subtrees $T_1, \ldots, T_k$, by first recursively drawing $T_1, \ldots, T_k$, and then by rearranging the subdrawings so to satisfy Invariants 1 and 2 (see Fig. 3(a)).

It is easy to see that the algorithm in Fig. 1 computes in linear-time a layered drawing on the yz-plane that satisfies both the previous two invariants. Moreover, the width (respectively, the height) of the drawing is at most n (respectively, n^2), where n is the number of nodes of the tree.

3.2 Phase 2: Equally Space the Children.

Let u be an internal vertex of T and $v_1, \ldots, v_k$ its children. In this phase we assign different x-coordinates to vertices $v_1, \ldots, v_k$ so that all edges (u, v_i), with $1 \leq i \leq k$, have the same length. Let $D(u)$ be the disk on the layer containing $v_1, \ldots, v_k$ and having as antipodal points v_1 and v_k (see Fig. 3(b)). We translate $v_2, \ldots, v_{k-1}$ along the x-direction until they meet the boundary of $D(u)$ (see Fig. 3(b)).

Algorithm `move(`T`)` in Fig. 2 implements the above strategy in linear time.

---

algorithm `move`(T)
$r \leftarrow$ root of T
$r_1, \ldots, r_k \leftarrow$ roots of $T_1, \ldots, T_k$ children of r
$d = d(r_1, r_k)$
for $i = 2$ **to** $k - 1$ **do begin**
    $x_{r_i} = x_r + \sqrt{d^2/4 - y_{r_i}^2}$
end
for $i = 1$ **to** k **do begin**
    `move`(T_{r_i})
end

Fig. 2. Phase 2: Algorithm `move`.

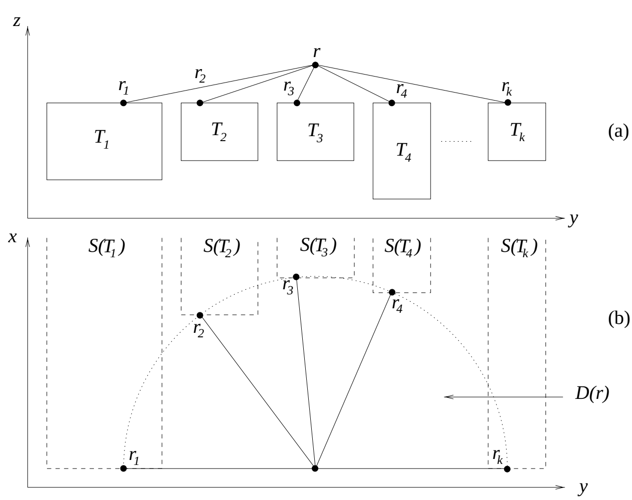

Fig. 3. (a) Phase 1: the front drawing of T. (b) Phase 2: how to equally space the children from the parent.

4 Proof of Correctness.

In this section we prove that the drawing obtained after Phase 2 is a weak Gabriel drawing for any tree T given in input. Moreover, its volume is n^4, where n equals the number of nodes of T. To this aim we first need some technical lemmas.

Lemma 1. *Let the distance δ of any two consecutive layers be at least n. For any two adjacent vertices a and b, $R[a,b]$ intersects only layers L_a and L_b.*

Proof. It is easy to see that the length of any edge is at most $\sqrt{n^2+n^2+n^2} = \sqrt{3}n$. On the counterpart the distance from the center of $R[a,b]$ to any layer other than L_a or L_b is $3n/2$, which implies that such a layer does not intersect $R[a,b]$.

Hereafter, we assume $\delta = n$.

Lemma 2. *Let a, b and $c \neq a, b$ be three points such that $d(a,b) = d(b,c)$. Then, $c \notin R[a,b]$.*

For any two points a and b on the same layer L, we denote by $D[a,b]$ the closed disk on L having a and b as antipodal points.

Lemma 3. *Let a and b be any two points and let b' be the projection of b on layer L_a. Then,*

$$R[a,b] \cap L_a = D[a,b'].$$

From now on, we identify a vertex r with the point of the space that represents r in the three-dimensional drawing of Phase 2. Additionally, we denote with $S(r)$ the semi-infinite strip of the space of minimal width containing all the vertices in the subtree rooted at r (see Fig. 3(b)).

Lemma 4. *Let u be any vertex and let $v_1, \ldots, v_k$ be its children. Then the following relations hold: (a) $R[u,v_i] \cap L_{v_1} \subset S(u)$. (b) $R[u,v_i] \cap L_u \subset S(u)$.*

Proof. We distinguish the two parts of the lemma.

(a) $R[u,v_i] \cap L_{v_1} \subset S(u)$.
Let us consider the disk $D(u)$ containing $v_1, \ldots, v_k$ (see Fig. 3). We apply Lemma 3 with $a = v_i$ and $b = u$. If we denote with u' the projection of u on L_{v_1} then we have:

$$R[u,v_i] \cap L_{v_1} = D[u',v_i] \subset D(u) \subset S(u).$$

(b) $R[u,v_i] \cap L_u \subset S(u)$.
Let $D'(u)$ be the disk $D(u)$ translated on L_u and centered at u. Also, let v_i' be the projection of v_i on L_u. By applying Lemma 3 with $a = u$ and $b = v_i$ we obtain:

$$R[u,v_i] \cap L_u = D[u,v_i'] \subset D'(u) \subset S(u).$$

The lemma thus follows.

We are now in a position to prove that the algorithm described in the previous section correctly constructs a weak Gabriel drawing for any tree T.

Lemma 5. *The drawing obtained after Phase 2 is a weak Gabriel drawing.*

Proof. We have to prove that, for any edge (u, v_i), the proximity region $R[u, v_i]$ does not contain any other vertex. Suppose, by contradiction, that a vertex v^* falls within $R[u, v_i]$. Then, from Lemma 1, only one of the following two cases is possible:

$v^* \in L_{v_i}$ (v^* *is on the same layer of* $v_1, \ldots, v_k$)
From Lemma 4, v^* must fall within $S(u)$. But the only vertices in $S(u) \cap L_{v_i}$ are the children of u. Thus, $d(u, v^*) = d(u, v_i)$. From Lemma 2, $v^* \notin L_{v_i}$, thus a contradiction.

$v^* \in L_u$ (v^* *is on the same layer of* u)
It is easy to see that it must be $S(v^*) \cap S(u) = \emptyset$ (see Fig. 3). From Lemma 4 we have that $v^* \notin R[u, v_i]$, thus a contradiction.

Let us now consider the size of the drawing.

Lemma 6. *The volume of the drawing obtained after Phase 2 is at most* n^4*.*

Proof. It is easy to see that the height is at most n^2, and the width is less than or equal to n. We have to prove that also the depth $depth(T)$ is at most n. The proof is by induction on n.

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

Inductive step. Let us suppose the lemma holds for trees with at most $n-1$ nodes, and let T be an n node tree. Let also $T_1, \ldots, T_k$ be its immediate subtrees, whose number of vertices are $n_1, \ldots, n_k$, respectively. Let also suppose T_1 be the larger immediate subtree. Then, from the algorithm `move` and considering that the initial width is at most n, we have that (see Fig. 3(b)):

$$\begin{aligned} depth(T) &\leq \max\{depth(T_1), n/2 + depth(T_2), \ldots, n/2 + depth(T_k)\} \\ &\leq \max\{n_1, n/2 + n_k\} \leq n, \end{aligned}$$

where the second last inequality follows from the inductive hypothesis, and the last one comes from $n_i \leq n/2$, for $2 \leq i \leq k$.

By combining Lemma 5 and Lemma 6 we get the following result.

Theorem 1. *Any tree admits a three-dimensional weak proximity drawing of volume at most* n^4*.*

The drawing produced by algorithms `front-drawing` and `move` requires a real RAM to represent vertex coordinates. Nevertheless, it is possible to slightly modify them to obtain proximity drawings with $O(\log n)$-bit requirement without increasing the volume.

First, notice that, by construction, all vertices have integer z-coordinate. We can modify algorithm `front-drawing` so that also the y-coordinate is an integer, by rounding it to the nearest integer value. Finally, the x-coordinate can be represented using the first $O(\log n)$-bits. It can be proved that the drawing so obtained is still a Gabriel drawing. Hence:

Theorem 2. *Any tree admits a three-dimensional weak proximity drawing of volume at most n^4 with $O(\log n)$-bit requirement.*

Finally, it is easy to see that if the input tree T is a binary tree then algorithm `front_drawing`, with $\delta = n/2 + 1$, produces a two–dimensional Gabriel drawing of T. Hence:

Theorem 3. *Any binary tree admits a two-dimensional weak proximity drawing with integer coordinates and at most n^3 area.*

5 Exponential Area versus Polynomial Volume.

In this section we describe an infinite class of graphs such that any Gabriel drawing requires exponential area, while they can be drawn in the three–dimensional space in linear volume, instead. This gives evidence that the use of the third dimension can substantially help in improving the efficiency and the effectiveness of the drawings. Additionally, we extend the results to β-drawings for $1 \leq \beta < 2$ and to relative neighborhood drawings.

The class has been introduced in [19], and in [16] the authors proved an exponential–area lower bound.

5.1 Class of Graphs.

The class is inductively defined as follows. Graph G_1 is the graph shown in Fig. 5(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. 5(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 form a triangle with a vertex of P_{i+1}, as well as each side of P_{i+1} with vertices in P_i. We refer such triangles as *petals*.

The main result of [16] is the following.

Theorem 4 ([16]). *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 same paper, the authors generalized the previous result to β-drawings, for any $1 \leq \beta < \frac{1}{1-\cos 2\pi/5}$.

5.2 Linear-Volume Drawings.

In this section we describe a linear-time algorithm to construct a *linear-volume strong Gabriel* drawing of G_n. The correctness of the algorithm will be proved in the next section.

We assume a *constant* distance δ between any two layers. The value of δ will be specified later.

Consider the algorithm `pentagons_in_3d` of Fig. 4. All pentagons P_i are equally drawn on different consecutive layers as regular pentagons and then rotated by a $\pi/5$ angle. Since the distance of consecutive layers is constant, and P_i is drawn in constant area, the volume is $O(n)$. Fig. 6(c) shows a drawing of G_4.

We will see in the following that, with a suitable choice of the value of δ, the drawing we obtain is a strong Gabriel drawing.

---

algorithm `pentagons`_in_3d(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 as P_{i-1} rotated by $\pi/5$
  connect P_i with P_{i-1}
end

Fig. 4. The algorithm to draw G_n in linear volume.

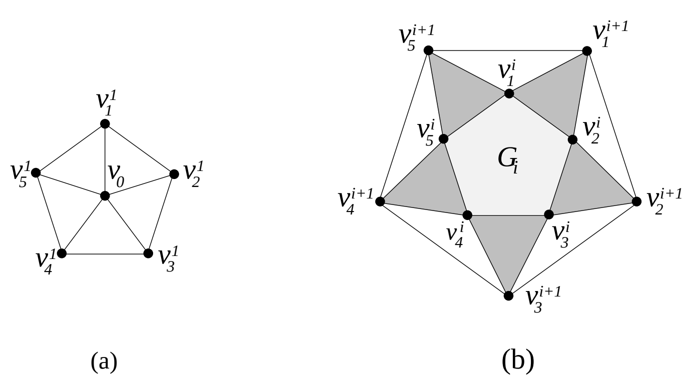

Fig. 5. The exponential-area/linear-volume class: (a) Graph G_1. (b) Graph G_{i+1} given G_i.

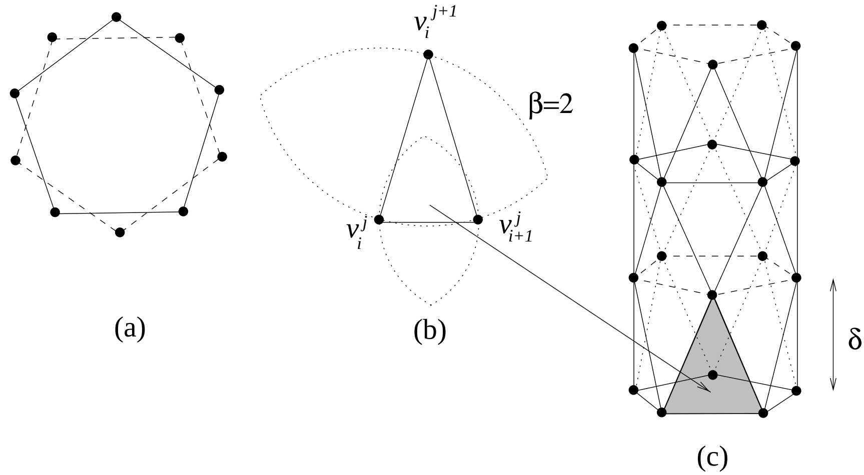

Fig. 6. (a) Two consecutive pentagons viewed from top. (b) How to draw a single petal. (c) The whole three dimensional drawing.

5.3 Proof of Correctness.

In this section we prove that the drawing obtained by algorithm `pentagons_in_3d` is a strong Gabriel drawing of G_n. To this aim, we have to prove that: (a) The proximity region of any two *adjacent vertices* does not contain any other vertex; (b) The proximity region of any two *non-adjacent vertices* contains at least another vertex. First note that G_1 is Gabriel drawable on layer 1. We distinguish the two cases:

Adjacent vertices We have the following two cases:

1. *The edge is a side of a pentagon.* Simply observe that no vertex of the pentagon itself can fall within the proximity region. Let l be the length of the side of a pentagon. Set δ at least equal to $l/2+1$. Then, no vertex of other pentagons falls within the region. Notice that the value of δ is proportional to l, and does not depend on the number of nodes of G_n.
2. *The edge connects two pentagons.* In this case the edge belongs to a petal. Since all the petals are equally drawn, then the vertex closest to the region is the opposite vertex of the petal itself. It is easy to see that such a vertex is outside the region, if the petals are drawn as isosceles triangles. (see Fig. 6(b)).

Non-adjacent vertices We distinguish the following two cases:

1. *The two vertices are not on consecutive layers.* There always exists a value of $\delta \geq l/2+1$ so that, for each vertex in P_i and each vertex in P_{i+2}, at least one vertex of P_{i+1} falls within the region of influence.

2. *The two vertices are on consecutive layers.* Without loss of generality, let us consider vertices v_1^i and v_2^{i+1}, and apply Lemma 3 with $a = v_1^i$ and $b = v_2^{i+1}$. If we denote with b' the projection of $b = v_2^{i+1}$ on layer L_a, we have that $v_1^{i+1} \in D[a, b']$ (see Fig. 6(a)). Thus, from Lemma 3, we have that $v_1^{i+1} \in R[a, b] = R[v_1^i, v_2^{i+1}]$. We can reasoning similarly in the other cases.

From the above discussion we derive the following fact.

Theorem 5. *For any n, graph G_n admits a three-dimensional strong Gabriel drawing of volume $O(n)$.*

Slightly modifying algorithm `pentagons_in_3d` is possible to extend Theorem 5 to strong β proximity drawings for $1 \leq \beta < 2$. In particular, it can be extended to produce relative neighborhood drawings of G_n. More precisely, for $1 \leq \beta < \frac{1}{1-\cos 2\pi/5}$ the algorithm is the same as in Fig. 4, with a suitable choice of δ. To obtain a relative neighborhood drawing of G_n we need to draw G_1 in a different way. Translate vertically vertex v_0 on level 0 leaving vertices of the pentagon P_1 on level 1 as in algorithm `pentagons_in_3d`. It easy to verify that with a suitable choice of δ the drawing so produced is a relative neighborhood drawing. Hence:

Theorem 6. *For any n, graph G_n admits a three-dimensional strong β-proximity drawing of volume $O(n)$, for any $1 \leq \beta < 2$.*

6 Open Problems.

Several problems concerning polynomial size proximity drawings are open. First of all it would be interesting to investigate possible extensions of our result along one or more of these directions:

1. *Extend β value.* Consider β-drawings of trees for $1 < \beta < 2$.
2. *Extend the class.* The first candidate might be outerplanar graphs.
3. *Strong proximity.* In particular, do at least binary trees admit polynomial volume strong proximity drawings?
4. *Two-dimensional drawings.* Can trees be drawn with polynomial area?.

More generally, it would be interesting to consider the area requirement of other proximity drawings such as minimum spanning tree or relative neighborhood drawings. Finally, another important research direction is to characterize classes of graphs which admit proximity drawing. Notice that for trees this problem is solved for several definitions of proximity (including Gabriel) both in two and three dimensions [4, 5, 15, 11].

Acknowledgments. The authors warmly thanks Prof. Roberto Tamassia and Dott. Giuseppe Liotta for their suggestions and helpful discussions. The problem tackled in this paper was raised and a preliminary study was done for the two–dimensional case while the second author was visiting the Center for Computational Geometry at Brown University.

References

[1] H. Alt, M. Godau, and S. Whitesides. Universal 3-dimensional visibility representations for graphs. In F. J. Brandenburg, editor, *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 8–19. Springer–Verlag, 1996.

[2] G. Di Battista, W. Lenhart, and G. Liotta. Proximity drawability : a survey. In *Proc. Graph Drawing '94*, Lecture Notes in Computer Science, pages 328–339. Springer Verlag, 1994.

[3] G. Di Battista, G. Liotta, and S.H. Whitesides. The strenght of weak proximity. In F. J. Brandenburg, editor, *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 178–189. Springer–Verlag, 1996.

[4] P. Bose, G. Di Battista, W. Lenhart, and G. Liotta. Proximity constraints and representable trees. In *Proc. Graph Drawing '94*, LNCS, pages 340–351. Springer–Verlag, 1994.

[5] P. Bose, W. Lenhart, and G. Liotta. Characterizing proximity trees. *Algorithmica*, 16:83–110, 1996.

[6] M. Brown and M. Najork. Algorithm animation using 3d interactive graphics. In *Proc. ACM Symp. on User Interface Software and Technology*, pages 93–100, 1993.

[7] T. Calamoneri and A. Sterbini. Drawing 2-, 3- and 4-colorable graphs in $o(n^2)$ volume. In Stephen North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 53–62. Springer–Verlag, 1997.

[8] M. Chrobak, M. T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proc. 12th Annu. ACM Sympos. Comput. Geom*, pages 319–328, 1996.

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

[10] P. Eades and S. Whitesides. The realization problem for euclidean minimum spanning tree is NP-hard. In *Proc. ACM Symp. on Computational Geometry*, pages 49–56, 1994.

[11] H. ElGindy, G. Liotta, A. Lubiw, H. Meijer, and S. H. Whitesides. Recognizing rectangle of influence drawable graphs. In *Proc. Graph Drawing '94*, number LNCS, pages 352–363. Springer–Verlag, 1994.

[12] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[13] A. Garg and R. Tamassia. GIOTTO3D: A system for visualizing hierarchical structures in 3d. In Stephen North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 193–200, 1997.

[14] D.G. Kirkpatrick and J.D. Radke. A framework for computational morphology. In G.T.Toussaint, editor, *Computational Geometry*, pages 217–248, Amsterdam, Netherlands, 1985. North–Holland.

[15] G. Liotta and G. Di Battista. Computing proximity drawings of trees in the 3–dimensional space. In *Proc. 4th Workshop Algorithms Data Struct*, volume 955 of *Lecture Notes in Computer Science*, pages 239–250. Springer–Verlag, 1995.

[16] G. Liotta, R. Tamassia, J. G. Tollis, and P. Vocca. Area requirement of gabriel drawings. In Giancarlo Bongiovanni, Daniel Pierre Bovet, and Giuseppe Di Battista, editors, *Proc. CIAC'97*, volume 1203 of *Lecture Notes in Computer Science*, pages 135–146. Spriger–Verlag, 1997.

[17] A. Lubiw and N. Sleumer. All maximal outerplanar graphs are relative neighborhood graphs. In *Proc. CCCG'93*, pages 198–203, 1993.

[18] J. MacKinley, G. Robertson, and S. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *Proc. of SIGCHI Conf. on Human Factors in Computing*, pages 189–194, 1991.

[19] D. W. Matula and R. R. Sokal. Properties of gabriel graphs relevant to geographic variation research and clustering of points in the plane. *Geogr. Anal.*, 12:205–222, 1980.

[20] M. Patrignani and F. Vargiu. 3DCube: A tool for three dimensional graph drawing. In Giseppe Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 284–290. Springer–Verlag, 1997.

[21] J.D. Radke. On the shape of a set of points. In G.T. Toussaint, editor, *Computational Morphology*, pages 105–136, Amsterdam, The Netherlands, 1988. North–Holland.

[22] S. Reiss. An engine for the 3D visualization of program information. *J. Visual Languages and Computing*, 6(3), 1995.

[23] G.T. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern recognition*, 12:261–268, 1980.