

# Polynomial Area Bounds for MST Embeddings of Trees

Michael Kaufmann

Wilhelm-Schickard-Institut für Informatik – Universität Tübingen, Germany

**Abstract.** In their seminal paper on geometric minimum spanning trees, Monma and Suri [6] gave a method to embed any tree of maximal degree 5 as a minimum spanning tree in the Euclidean plane. They derived area bounds of  $O(2^{k^2} \times 2^{k^2})$  for trees of height  $k$  and conjectured that an improvement below  $c^n \times c^n$  is not possible for some constant  $c > 0$ . We partially disprove this conjecture by giving polynomial area bounds for arbitrary trees of maximal degree 3 and 4.

## 1 Introduction

As a classical concept, a minimum spanning tree (MST) of  $n$  points in the plane is defined as being the smallest tree that spans all  $n$  points. As the distance between two non-adjacent vertices must be at least as large as any distance between adjacent vertices on the path between the two vertices, the MST reflects certain proximity relations between the points in the plane, it plays important roles in various fields. It recently gained attention in the field of sensor networks where the positioning and energy consumption of sensor nodes is an important issue. For example, in [4] the authors use the FIRST ORDER RADIO MODEL as energy model, to develop heuristics for MSTs in the plane with hop and degree constraints.

We reconsider the embedding question: Given a certain tree topology as an input, find geometric positions in the Euclidean plane for the vertices of the tree, such that the geometric MST for those points exactly corresponds to the given tree topology. We call this problem the MST embedding problem. Monma and Suri [6] gave a method to find such positions for trees of maximal degree 5, and they showed that their algorithm uses an area of  $O(2^{k^2} \times 2^{k^2})$  for trees of height  $k$  and made a simple observation, that trees of degree 7 cannot be drawn as Euclidean MST. Eades and Whitesides [3] filled the gap by their result that the decision whether an MST embedding is possible for a given tree of degree 6 is NP-hard. In both papers, the open problem is given whether or not the area bounds can be improved to polynomial. Note that the algorithm of Monma and Suri does not even give a polynomial bound for complete trees of maximal degree 5, their area bound there reads  $n^{O(\log n)}$ , which is clearly superpolynomial.

Extensions of this work to 3D have been performed by Di Battista and Liotta [2] as well as by King [5]. In the former paper, the authors proved that trees with maximal degree 9 can be embedded as MSTs in three-dimensional space; this result has been recently extended to trees of maximal degree 10.

In this paper, we concentrate on the area requirements for Euclidean MST embeddings. First, we give a simple technique for complete binary trees, which will be extended to the case of arbitrary binary trees. In the second part of the paper, we consider

ternary trees, which are trees of maximum degree 4. Again, we develop a basic scheme for the complete ternary tree, and generalize it afterwards to arbitrary ternary trees.

Note that we do not strive for the best polynomial bounds but try to keep the techniques as simple as possible. Nevertheless, we achieve the first polynomial bounds drastically improving from the previous exponential ones and solving long-standing open problems.

## 2 Preliminaries

A minimum spanning tree  $MST$  of a set of  $n$  points in the plane is defined to be a tree with lowest total cost where the cost of each edge  $(u, v)$  is defined by the Euclidean distance between  $u$  and  $v$ . Given a tree  $T$ , the MST embedding problem now asks for a mapping of the vertices in  $T$  to points in the plane such that the MST of this point set is exactly the input tree  $T$ . This problem can be solved for trees of maximum degree 5, it cannot be solved for trees of maximum degree larger than 6, and it is NP-hard to decide whether there exists a MST embedding for a given tree of maximum degree 6. Furthermore it is known that Euclidean MSTs have the properties that all edges on the path in the tree between non-adjacent vertices  $v$  and  $w$  are not longer than the distance between  $v$  and  $w$  in the plane. We call this the MST condition. In our constructions, we have to prove that the MST condition has been observed. It is clear that the edges are non-crossing when drawn straight-line. We define the area consumption of an embedding to be the area of the rectangle enclosing the embedding.

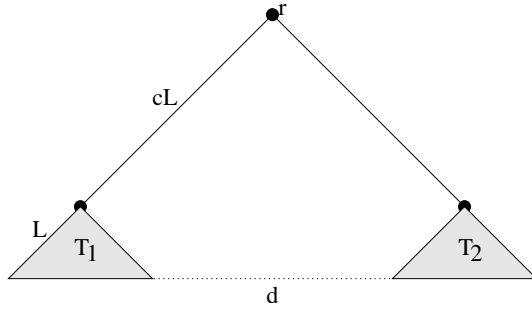
## 3 The Complete Binary Case

Let  $T$  be a complete binary tree of size  $n$  and let  $n = 2^k - 1$  for some integer  $k$ . It consists of a root vertex  $r$  and two subtrees  $T_1$  and  $T_2$  rooted at the direct children  $r_1$  and  $r_2$  of the root. We recursively embed the subtrees of size  $n/2$  into two equal-sized cones with angle of 90 degrees where the roots are located at the top of the cones of the corresponding subtree. We place the two cones of the subtrees next to each other separated by a distance of  $d$  where  $d$  will be chosen later. Let  $L$  denote the side length of the cones of the subtrees. Then we place the root  $r$  on top of the two cones connecting the two roots by edges of length  $c \cdot L$  such that those edges again form an angle of 90 degrees. Clearly, the whole construction again gives a cone of side length  $(c + 1) \cdot L$  where the root  $r$  is placed at the top, cf. Figure 1.

By the construction, the distance  $d$  between the inner corners of the cones is  $d = \sqrt{2}(c - 1)L$ . To ensure the conditions for the MST, we choose  $c$  such that  $d$  is larger than the length  $cL$  of the two longest edges which is incident to the root. All other distances obey the MST conditions by induction. This is true if  $c \geq \sqrt{2}/(\sqrt{2} - 1)$ .

Hence we obtain a recursion for the sidelength  $S(n) = (c + 1)S(n/2) = (\frac{2\sqrt{2}-1}{\sqrt{2}-1})^{\log n}$  which is  $n^{\log \frac{2\sqrt{2}-1}{\sqrt{2}-1}} \leq n^{\log 4.415} \leq n^{2.2}$ . From this, we obtain an area bound of  $O(n^{4.4})$ .

**Theorem 1.** *We can embed a complete binary tree of  $n$  vertices in an area of size  $O(n^{4.4})$  such that the embedding obeys the MST conditions.*



**Fig. 1.** The recursive construction for the complete binary case. Note that the figure should be seen as a sketch, the actual values in the figure violate the MST condition.

## 4 The Arbitrary Binary Case

This case is more involved. Before, our analysis relied on the fact that the depth of the tree and hence the recursion had only depth  $\log n$ , which results in a polynomial bound for the area. The basic schema remains the same. We recursively embed the subtrees into cones with uniform angles of 90 degrees and try to combine two subtrees with two sibling vertices as roots by placing the common ancestor such that they form another larger cone observing the necessary distances. The two subtrees and hence their cones might differ more or less in size. If their cones differ by at most a constant factor each time, we can use an analogous construction as before to achieve a depth still logarithmic in  $n$ . If not, we have to introduce an alternative technique.

More exactly, let  $P = v_1, v_2, \dots, v_k$  be the path from root  $v_1$  to leaf  $v_k$  where  $v_{i+1}$  is defined as the root of the larger of the two subtrees of  $v_i$  for each  $1 \leq i < k$ . For each such  $v_i$  on this path, we consider the two subtrees  $T_{il}$  and  $T_{ir}$ , and we assume that  $T_{il}$  is at least as big as  $T_{ir}$ . If  $|T_{il}|/|T_{ir}| \leq 3$  we call this a balanced step, and the step is called unbalanced otherwise. Clearly, there are at most  $O(\log n)$  balanced steps. Between any two balanced steps there is a sequence  $v_f, \dots, v_g$  of unbalanced steps, of course there are at most  $O(\log n)$  such sequences.

**Realization of a balanced step at the vertex  $v_i$  with subtrees  $T_{il}$  and  $T_{ir}$ .** Let  $L_l$  and  $L_r$  be the lengths of the two cones where the subtrees  $T_{il}$  and  $T_{ir}$  are embedded and let  $L_l \geq L_r$ .

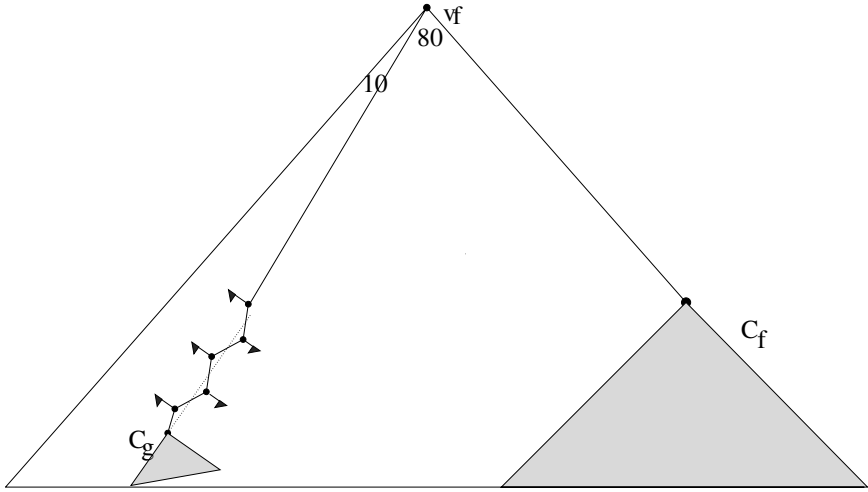
We scale the smaller cone up by a factor of  $L_l/L_r$  such that the two cones have the same size and the same analysis as in the complete binary case can be done. Since the number of nodes in the larger subtree  $T_{il}$  is at most  $3/4$  times the number of nodes in the original tree rooted at  $v_i$  this case can occur only  $\log_{4/3} n$  times.

**Realization of a maximal sequence of unbalanced steps.** Let the sequence be  $v_f, \dots, v_g$  with intermediate edges  $(v_i, v_{i+1})$  for  $i = f, \dots, g-1$ . This sequence with the edges is called *chain*. The final node  $v_g$  is either a leaf or it is the root of two balanced subtrees. We assume that the subtree rooted at  $v_g$  has been recursively embedded into

cone  $C_g$  by a balanced step. At each intermediate vertex  $v_i$ , there will be a relatively small subtree, say  $T_{ir}$  embedded recursively into cone  $C_i$  of side length  $L_i$ .

If the sequence is very short ( $g - f \leq 2$ ), say only one or two unbalanced steps, followed by a balanced one, we will just neglect it and perform one or two ordinary balanced steps instead as described above. If  $g - f > 2$  is odd, we perform the layout of the topmost two trees of the chain by balanced steps, while if  $g - f > 2$  is even, we perform only one balanced step and then add the remaining unbalanced steps of the chain in one step. This is done to make sure that the last subtree of the chain with unbalanced layout lies towards the left of the chain.

It is clear, that the number of balanced recursion steps increases only by a factor of at most 3, so it is still  $O(\log n)$ , more exactly the number is at most  $3 \log_{4/3} n$ . We will use this fact again by correcting the following construction:



**Fig. 2.** The global recursive structure when adding a whole chain and the cone  $C_f$

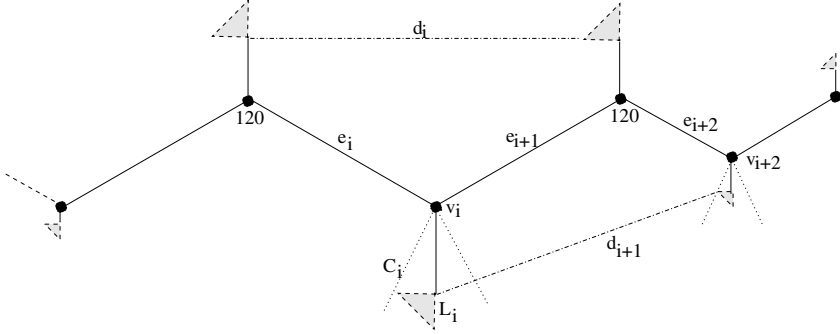
Let  $e_i = (v_{i-1}, v_i)$  for  $g \geq i \geq f$ . We embed the chain of edges  $e_i$  in a zig-zag way with constant angles of 120 degrees, while the length of  $e_i$  will be chosen roughly proportional to the size of the adjacent cone. Consider an arbitrary vertex  $v_i$ . Opposite to the 120 degree angle, we have an angle of 240 degrees, which we subdivide into three disjoint cones of 90, 60, and 90 degrees. We place the cone  $C_i$  adjacent to  $v_i$  of size  $L_i$  together with its connecting segment  $s_i$  in the middle angle of 60 degrees such that the side of  $C_i$  which is closer to  $e_i$  is perpendicular to segment  $s_i$ . Furthermore we choose the length of  $s_i$  such that the small cone just fits into the angle of 60 degrees. Hence  $s_i = L_i / \tan(30) \leq 1.8L_i$ .

Note that by construction the MST condition is observed for the distances between  $C_i$  and  $C_j$  for  $|i - j|$  odd, namely between cones that lie on opposite sides of the chain.

We consider the horizontal distance between two adjacent cones  $C_{i-1}$  and  $C_{i+1}$ . The intermediate edges are  $s_{i-1}, e_i, e_{i+1}$  and  $s_{i+1}$ . Clearly we have distance  $d_i = \text{len}(e_i) \cdot \cos(30) + \text{len}(e_{i+1}) \cdot \cos(30) - L_{i+1} \geq 0.8c(L_i + L_{i+1}) - L_{i+1}$ , if we

choose  $\text{len}(e_i) = cL_i$ . Now to observe the MST condition, we have to choose  $c$  such that  $d_i \geq \text{len}(e_i)$ , and  $d_i \geq \text{len}(e_{i+1})$  as well. Therefore, we require that  $\text{len}(e_i) \geq \max\{cL_i, 0.5\text{len}(e_{i+1})\}$  and  $\text{len}(e_{i+1}) \geq 0.5\text{len}(e_i)$  as well for all  $i$ .

Then we can conclude that  $d_i \geq 0.8 \cdot 1.5 \cdot cL_{i+1} - L_{i+1} \geq cL_{i+1}$  for  $c \geq 5$ . If  $\text{len}(e_i) > 0.5\text{len}(e_{i+1})$  then  $c$  can be chosen even smaller. The case that  $\text{len}(e_i) \geq \text{len}(e_{i+1})$  is simpler.



**Fig. 3.** Adding a chain of unbalanced vertices together with their adjacent subtrees

Note that segment  $s_{i+1}$  is smaller than  $\text{len}(e_{i+1})$ , and hence it is smaller than  $d_i$ , and  $s_{i-1}$  is smaller than  $1.8L_{i-1}$ . Furthermore  $\text{len}(e_{i-1}) \geq 5L_{i-1}$ , which means that  $\text{len}(e_i) \geq 0.5\text{len}(e_{i-1}) \geq 2.5L_{i-1} \geq \text{len}(s_{i-1})$ .

**Lemma 1.** *The distances between the vertices within the chain augmented by the adjacent subtrees observe the MST conditions.*

Ideally, the length of the chain should be bounded proportional to the sum of the lengths of the adjacent small cones. The requirement that the lengths of adjacent edges should differ by at most factor of 2 makes an estimation more difficult, but the following amortisation argument shows that this increases the total edge length of the chain at most by a factor of 3:  $e_i$  is chosen to be of length  $\max\{c \cdot L_i, c \cdot L_{i-1}/2, c \cdot L_{i+1}/2, \dots, c \cdot L_{i+j}/2^j\}$ . Hence each  $L_i$  at most contributes to the segment  $e_i$ , with half of its length to  $e_{i-1}$  and  $e_{i+1}$  and with  $1/2^j$ th of its length to  $e_{i-1}$  and  $e_{i+1}$  in general. Overall it is accounted at most three times. So, all  $L_i$ 's together are accounted at most  $3 \sum L_i$ .

**Lemma 2.** *The total length of the chain with cones  $C_i$  of length  $L_i$  is at most  $15 \sum_{i=f+1}^g L_i$ .*

Next, we sketch the way how to complete the recursion and achieve again an appropriate cone of 90 degrees.

**Completing the recursion step.** The next Figure 4 gives an intuition how to rebuild the giant cone including the chain of small cones and the first cone  $C_f$  as well: We add the first cone  $C_f$  of the chain in a way similar to a balanced step. Since the construction of the chain require some width, basically, we keep the angle between the two edges

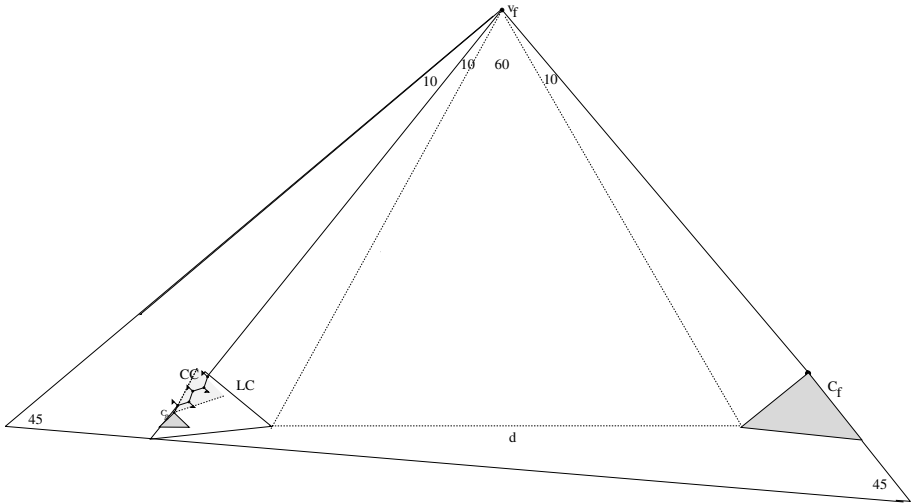
incident to the root of the two subtrees a little smaller than 90 degrees, namely 80 degrees. This leads to an increase of the lengths of the two topmost edges by a small constant factor to keep the necessary distance between the left and the right subtree. We keep the triangle between the root  $v_f$  and the two innermost points between the left and the right subtree to be equilateral. An equilateral triangle has angles of 60 degrees, hence we assign the remaining 20 degrees to the two subtrees, such that each of them lies within a cone of 10 degrees.

The edge  $e_f$  which connects  $v_f$  and  $v_{f+1}$  is routed using the same slope, as the left side of the cone  $C_g$  which is constructed recursively by a balanced step. The whole chain lies within a cone, called  $CC$  of 60 degrees whose middle axis extends the left side of  $C_g$ . The remaining 10 degrees of the final 90 degree cone are used to cover the small subtrees that stick out to the left of the line between the left side of the large cone  $C_g$  and the root  $v_f$ . We can easily verify that this small angle of 10 degrees is enough to include the whole substructure by observing the restrictions on the size of the small subtrees and the sufficient lengths of the edges from the chain to the root.

We have to embed all three parts, the cone  $C_f$ , the chain plus the cone  $C_g$ , and the small cones like  $C_{f+1}$  sticking out to the left, within cones of 10 degrees each. The final size of the giant cone depends on the actual sizes of the three components, at the end, we maximize over those three components. We give the analysis for the case that the chain dominates the final size, assuming that  $C_f$  is small and the other small cones are also small enough. The other cases are much simpler:

The length of the chain is at most  $15 \sum_{i=f+1}^g L_i$ , we call it  $LC$ . Largely overestimating, we require this value also for the cathetus opposite from the 10 degrees angle at  $v_f$ . Hence we get a bound of  $LC/\tan(10) \leq 5.7LC$  for the length of  $(v_f, v_{f+1})$ .

Note that this can be slightly larger, if the chain starts at the upper end of its upward directed cone  $CC$ , but not in the center. Furthermore we neglected the fact, that due to



**Fig. 4.** The giant cone which is rotated such that the three parts  $C_f$ , the chain and the leftmost small cones are enclosed into cones of angles of 10 degrees, and the equilateral triangle keeps the distance

the separating equilateral triangle and the additional angle of 10 degrees on the left hand side, the sidelength of the final giant cone is larger than  $LC + 5.7LC$  by an additional small factor. This factor can be bounded using more elementary trigonometry by 1.16.

Hence the new giant cone has a side length of  $1.16 \cdot (5.7 + 1)15 \sum_{i=f}^g L_i \leq 117 \sum_{i=f}^g L_i$ . Summarizing, we have

**Lemma 3.** *Adding a chain causes an increase of the side length, which is proportional to the sum of the lengths of its subtrees, with 117 as the multiplicative constant.*

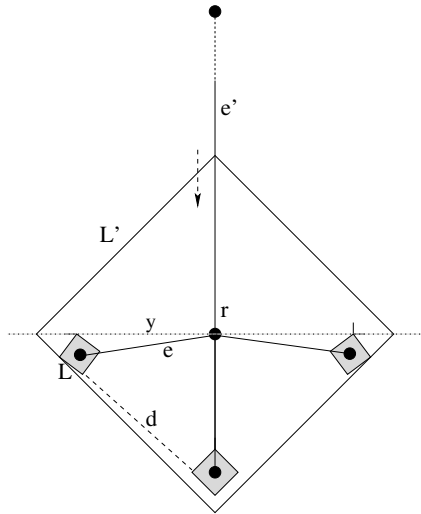
We conclude the discussion by the recursion on the side length  $S(n)$ : We had at most  $\log_{4/3} n$  balanced steps, and before each balanced step, there might be the insertion of a maximal chain consisting of unbalanced steps, hence only  $\log_{4/3} n$  subsequent insertions of chains. In total, this means a maximal recursion depth of  $2 \log_{4/3} n$ , we get  $S(n) \leq 117^{2 \log_{4/3} n} \leq O(n^{32.4})$ .

**Theorem 2.** *We can embed an arbitrary binary tree of  $n$  vertices in an area of size polynomial in  $n$  such that the embedding obeys the MST conditions. The degree of the polynomial is at most 65.*

Note that this analysis above is a rough estimation. E.g., in balanced steps we have a constant factor, much smaller than 117, and in unbalanced steps and long chains, the larger subtrees are much smaller than  $3n/4$ . We leave this for further improvement.

## 5 The Complete Ternary Case

Instead of maintaining cones for the subtrees, we maintain diamonds. Consider proper subtree  $T_r$  rooted at vertex  $r$ . Vertex  $r$  is placed in the centre of its corresponding diamond of side length  $L$ . The diamond for  $T_r$  is partitioned into an upper and a lower



**Fig. 5.** The recursive scheme for complete ternary trees

half. The lower half contains the diamonds for the three subtrees, such that the edges from  $r$  to the diamonds maximize the angles between two adjacent edges.

This is a simple recursive scheme and we only have to choose the lengths of the edges appropriately such that the MST property still holds. Furthermore, we have to consider the ratio between the long edge  $e'$  from  $r$  to its parent and the shorter edges  $e$  to its children. It will then provide the area bound.

Now let the side length of the small diamonds be  $L$ . The length of edge  $e$  is  $\text{len}(e)$  while the distance from vertex  $v$  to the upper / lower corners of the left and right diamonds is denoted by  $y$ . Note that  $y$  is larger than  $\text{len}(e)$ .

Clearly, the lower corner of the say left diamond and the left corner of the middle diamond should have a distance  $d$  of at least  $\text{len}(e)$ . Since  $y \geq \text{len}(e)$  it is sufficient to ensure that  $d = y$ . The key observation is that under this condition the vertex  $r$  together with the lower corner of the left diamond and the left corner of the middle diamond forms a equal-sided triangle. All angles in this triangle have 60 degrees. Hence, the three small angles between the edges from  $r$  and the lines from  $r$  to the corners of the two diamonds complete the right angle, hence they have 10 degrees. So, we can use more trigonometry:  $\tan(10) = (L/\sqrt{2})/\text{len}(e)$ . Hence  $\text{len}(e) = (L/\sqrt{2})/\tan(10) < 4.1L$ .

Furthermore, the distance from the parent of  $r$  to the upper corners of the diamonds on the side is larger than the length of the edge  $e' = (p, r)$ . Next, we compute the side length  $L'$  of the large diamond which encloses  $r$  and the three small diamonds. More exactly, we will compute the length of the horizontal axis through vertex  $r$ . We know from the construction before that the segment which extends the edge  $e$  from the centre of the diamond to the left end has length  $4.1L + L/\sqrt{2}$ . The angle between this segment and the horizontal line has 10 degrees. Hence the cathetus opposite of this angle has length  $\sin(10) \cdot (4.1L + L/\sqrt{2}) < 0.84L$ , we call it  $z$ . The length of the horizontal axis from  $r$  to the left corner is  $z + z/\tan(10) \leq 5.6L$ , since it consists of two subsequent parts.

From this consideration, we know the ratio between subsequent diamonds, and hence we can conclude the needed side length  $S(n) \leq 5.6S(n/3) = 5.6^{\log_3 n} \leq n^{3.94}$  and hence the area is polynomial in  $n$ .

**Theorem 3.** *We can embed a complete ternary tree of  $n$  vertices in an area of size  $O(n^{7.88})$  such that the embedding obeys the MST conditions.*

## 6 The Case of Arbitrary Ternary Trees

As in the binary case, we have to consider the case that the subtrees of the vertices might have different sizes, leading to a much larger depth than in the complete ternary case. Note that the case of binary trees might have been omitted after all, since it is just a subcase of the ternary case. The techniques used are similar to those for the general binary case, but they are even more involved. Therefore we omit the calculation of concrete exponents of the polynomials.

Consider a vertex  $v$  root of a subtree  $T$  of size  $n'$  divided into three subtrees  $T_1, T_2$  and  $T_3$  of sizes  $n_1, n_2$  and  $n_3$  respectively, such that  $n_1 \geq n_2 \geq n_3 \geq 0$ .  $T_i$  is called 'large' if  $n_i \geq n'/4$  and 'small' otherwise, for  $i = 1, 2, 3$ .



We consider 2 cases: If we have at least 2 large subtrees, we proceed similar to the complete ternary case. Observe that if we would be in this case each time, then the maximal size of a subtree would shrink in each iteration at least by a constant fraction  $3/4$  or so. area if we only follow the scheme as in the complete ternary case. This corresponds to the balanced recursion step in the binary case. We take the size of the three diamonds identical by scaling up the sizes of the edge lengths in the smaller subtrees. Since the recursion depth remains  $O(\log n)$ , the area size will still be polynomial.

The second case is much more interesting: We assume to have 2 small subtrees. We walk down into the large subtree and check, if there are at least 2 large subtrees starting from there. If this is true for either this step or the next step, such that there are again at least 2 large subtrees, we can neglect the area loss of the 2 small subtrees by scaling them up as before and make a balanced recursion step. Hence the number of large recursion steps at most triples by this effect but it still remains logarithmic.

**Constructing a chain of small diamond twins.** If not, we construct a maximal chain of vertices  $v_1, \dots, v_k$  with small subtrees represented by small diamonds. The indexing of the chain starts at the lowest vertex  $v_1$ , which actually has one large subtree and two small subtrees. In general,  $v_i$  has two small and one large subtree with root  $v_{i+1}$ , which again has one large subtree and two small subtrees. The vertex  $v_k$  is either the root, or its parent has two large subtrees, such that we can apply a balanced recursion step.

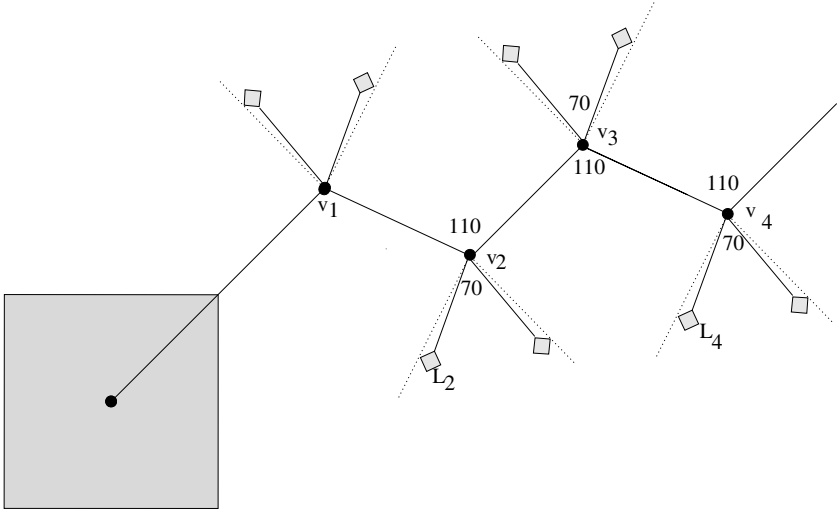


Fig. 6. Embedding a chain of small diamonds

To embed the chain with the twin diamonds appended at each vertex we take an approach similar to the general binary case: We embed the chain in a zig-zag way with angles of  $110$  degrees and the lengths are suitably chosen roughly proportional to the size of the adjacent diamonds. We will determine the exact edge lengths lateron.

Consider vertex  $v_i$ . Opposite of the  $110$  degree angle, we install a cone of  $70$  degrees where we place the two small diamonds adjacent to  $v_i$ . First of all, we scale the smaller

diamond such that both diamonds have the same size. The diamonds are connected with edges  $g_i$  and  $g'_i$  and they are placed tangent to the two sides of the cone. The lengths of the edges have to be chosen such that the distance  $d_i$  between the two diamonds is at least as large as the lengths of  $g_i$  and  $g'_i$ . We ensure this by enlarging the lengths such that together with  $v_i$  the two corners of the diamonds opposite to each other form an equilateral triangle. Hence the two diamonds have to be placed within the cones of degree 5 on the sides of the equilateral triangle. Solving  $\tan(2.5) = L_i/(\sqrt{2}\text{len}(g_i))$  where  $L_i$  is the width of the largest of the two diamonds, and  $\text{len}(g_i)$  gives the lengths of the two connecting edges. Hence  $\text{len}(g_i) = L_i/(\sqrt{2}\tan(2.5)) \approx 16.2 \cdot L_i$ .

Note that by the choice of the cone for the two diamonds, they are farer away from the position of  $v_{i-1}$  resp.  $v_{i+1}$  independent of the lengths of  $e_{i-1}$  and  $e_i$  in the chain.

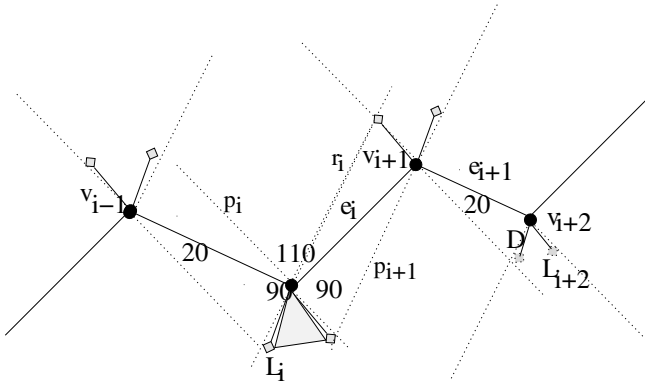
**The case of uniform sizes.** To actually choose the length of the edges  $e_i$ , assume for a moment, that we have uniform side lengths  $L_i$  of the diamonds. Then we can choose the lengths of the  $g_i$ 's and those of the  $e_i$ 's as being uniform. To keep the distance between the diamonds at  $v_i$  and  $v_{i+2}$  large enough, we consider the triangle with cathetus  $e_{i+1}$  and angle of 20 degrees at vertex  $v_{i+1}$ , and call the other cathetus  $D$ . If  $D \geq \text{len}(g_{i+2}) + \sqrt{2}L_{i+2} \geq 17.5L_{i+2}$ , the distance between the two diamonds is large enough. So,  $D = \tan(20) \cdot \text{len}(e_{i+1}) \geq 0.37\text{len}(e_{i+1})$ , and furthermore  $\text{len}(e_{i+1}) = 17.5 \cdot L_{i+2}/0.37 \geq 47.3L_{i+2}$ .

**The general case.** Unfortunately, this is not sufficient since we want to have the total length of the chain to be proportional to the sum of the  $L_i$ 's. So we have to observe the different sizes of the diamonds. We introduce another restrictive piece of construction.

The following calculation shows how much we can reduce the edge length within one step. Consider vertices  $v_i, v_{i+1}$  and  $v_{i+2}$  with connecting edges  $e_i$  and  $e_{i+1}$ . We choose  $\text{len}(e_i) = 47.3L_i$ , and want to reduce the length  $\text{len}(e_{i+1})$  of the next edge to  $3/4 \cdot \text{len}(e_i)$ . Let  $D$  be the cathetus opposite of the 20 degrees angle at  $v_{i+1}$ , where the small diamonds at  $v_{i+2}$  should be placed. Then  $\text{len}(D) = \text{len}(e_{i+1} \cdot \tan(20)) \geq 0.36\text{len}(e_{i+1}) = 0.36 \cdot 3/4 \cdot \text{len}(e_i) \geq \text{len}(e_i)/4$ . By this 20 degree angle, we introduced a line  $r_{i+1}$  parallel to the right border of the cone at  $v_i$  at the distance of  $\text{len}(e_{i+1})$ , keeping the diamonds incident to  $v_i$  and  $v_{i+2}$  respectively sufficiently apart from each other. Hence it is safe to embed at vertex  $v_{i+2}$  diamonds of size at most  $L_{i+2} = L_i/4$ , and the connecting edges having length  $\text{len}(g_{i+2}) = 17.5L_{i+2}$  while reducing the length of  $e_{i+1}$  by factor  $3/4$ .

In general, we introduce lines  $r_i$  and  $p_i$  at each vertex  $v_i$  parallel to the cone boundaries at the neighboring vertices  $v_{i-1}$  and  $v_{i+1}$ , extending the two lines of the cone at  $v_i$  to the other side of the chain. We require that the small diamonds adjacent to  $v_i$  and more specifically the length of  $g_i$  are restricted by the lines  $r_{i-1}$  and  $p_{i+1}$ , which is critical especially if the lengths of the edges  $e_i$  change. The above calculation showed to which extend edge  $e_{i+1}$  can be shortened if  $L_{i+2} \leq L_i/4$ . But note that the line  $p_{i+2}$  comes then closer to the edge  $e_{i+1}$  such that it might intersect the diamonds at  $v_{i+1}$ . This problem can be resolved by requiring that the edge  $e_{i+1}$  can only be shorted if the diamonds at both  $v_{i+1}$  and  $v_{i+2}$  as well are smaller by at least a factor of 4 compared to the diamonds at  $v_i$ .

We conclude the analysis and summarize the arguments for the correctness.



**Fig. 7.** A detailed look to the chain of small diamonds when the size of the diamonds and the length of the edges shrink. Notice the lines  $p_i$  and  $r_i$  parallel to the sides of the cones.

**Lemma 4.** *Proceeding from  $v_i$  to  $v_{i+2}$ , we can shrink the length of the second edge  $e_{i+1}$  by a factor of  $4/3$  compared to  $e_i$ , if the widths of the diamonds at  $v_{i+1}$  and  $v_{i+2}$  shrink by at least a factor of 4.*

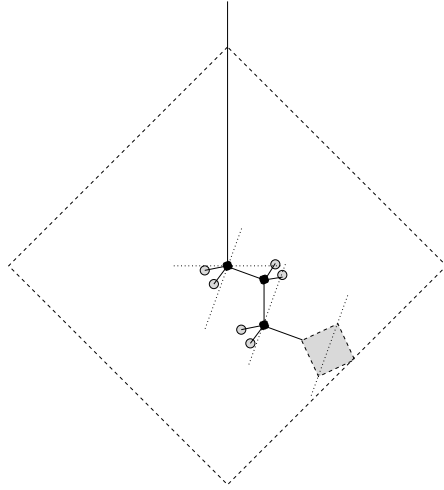
1. The Scaling up procedures do not violate any MST condition.
2. Cones at adjac. vertices do not conflict. They are strictly separated by construction.
3. Diamonds within the same cone do not interfere since they are separated by an equilateral triangle.
4. Diamonds adjacent to  $v_i$  do not interfere with those adjacent to  $v_{i+2}$ ; here we can argue using the separating lines parallel to the sides of the cone.

**The final length of the chain.** Note that we did not fix the length of  $e_i$ , its lower bound was  $47.3L_i$ , according to the lemma, the length depends on the size of the diamonds in the close or further vicinity. So we do the same with  $v_{i-1}$  and  $v_{i+1}$  and then maximize.

More precisely, for each  $v_i$ , we receive a lower bound of  $47.3L_i$  for the two incident edges  $e_{i-1}$  and  $e_i$  respectively. Furthermore, we have the dependency between neighboring edges  $e_i$  and  $e_{i+1}$ , namely that the ratio of the lengths can be chosen to be either  $3/4$ ,  $1$  or  $4/3$ . Remark that we could also choose some intermediate value, but for the clarity of exposition we take only those discrete values.

Finally, we argue that the sum over the lengths of the edges  $e_i$  on the chain is polynomial in the number of vertices involved (more exactly in the sum over the lengths of all small diamonds). We have to cover the diamonds involved with a sequence of larger diamonds of a certain size such that adjacent groups might either be identical or differ by a ratio of  $1/4$  or  $4$ . In these cases, the edge lengths stay either identical, shrink or enlarge by a factor of  $4/3$  respectively.

By an amortization argument which is not much more complicated than the one above, we can show that the sum of the edge lengths is still  $O(L)$ , where  $L = \sum_{i=1}^k L_i$ , hence linear in the sum of the side lengths of the diamonds involved.



**Fig. 8.** Placing the chain of small diamonds plus the large diamond within the giant diamond

**The recursive construction and its complexity.** Finally, we use the construction from Figure 8 to combine the last large diamond of size  $n_0$  with the chain of diamonds of sizes  $n_1, \dots, n_k$  and get the following recursion for the resulting side length:

$S(n) = c_0 S(n_0) + c_1 \cdot \sum_{i=1}^k S(n_i)$  for appropriate constants  $c_0$  and  $c_1$ . Since we have at most  $O(\log n)$  levels of recursion while  $n = \sum_{i=0}^k n_i$  and  $n_i \leq 3/4n$  for any  $i = 0, \dots, k$  we conclude in the same way as in the general binary case:

**Theorem 4.** *Let  $T$  be any tree with  $n$  vertices of maximal degree 4. We can find an MST-embedding for  $T$  in a grid of size  $O(n^d)$  for a suitable large constant  $d$ .*

## 7 Discussion and Conclusion

We have shown some techniques for constructing and analysing area-efficient MST-embeddings of trees. Our goal was to prove polynomial area bounds improving the previously known exponential bounds. A more detailed analysis of course could provide exact constants for the exponents. We started with binary trees and some basic analysis. The complexity of the analysis increased when proceeding to ternary trees where the vertices have maximum degree 4. It is known that trees with maximum degree 5 also have MST-embeddings. For this case we expect an exponential area lower bound.

**Acknowledgement.** This work has been initiated at the Workshop of Graph Drawing and Computational Geometry in Bertinoro, Italy. Thanks to the organizers and participants for the inspiring atmosphere. The initial part about complete binary trees was done together with Roberto Tamassia. Special thanks also go to Fabrizio Frati, Markus Geyer and Barbara Pampel for many critical remarks und suggestions.

## References

1. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, Englewood Cliffs (1999)
2. Di Battista, G., Liotta, G.: Computing proximity drawings of trees in the 3-dimensionanl space. In: Sack, J.-R., Akl, S.G., Dehne, F., Santoro, N. (eds.) *WADS 1995*. LNCS, vol. 955, pp. 239–250. Springer, Heidelberg (1995)
3. Eades, P., Whitesides, S.: The Realization Problem for Euclidean Minimum Spanning Trees in NP-Hard. *Algorithmica* 16(1), 60–82, 1–15 (1996)
4. Cheng, H., Liu, Q., Jia, X.: Heuristic algorithms for real-time data aggregation in wireless sensor networks. In: *Proceedings of the 2006 International Conference on Communications and Mobile Computing* (2006)
5. King, J.: Realization of Degree 10 Minimum Spanning Trees in 3-Space. In: *CCCG 2006*. *Proceedings of the 18th Canadian Conference on Computational Geometry*, pp. 39–42 (2006)
6. Clyde, L., Monma, C.L., Suri, S.: Transitions in Geometric Minimum Spanning Trees. *Discrete & Computational Geometry* 8, 265–293 (1992)