Computing an Optimal Layout for Cone Trees

Dirk Zeckzer[1], Fang Chen[1], and Hans Hagen[1]

1 **Department of Computer Science**
University of Kaiserslautern
Kaiserslautern, Germany
{zeckzer,chen,hagen}@informatik.uni-kl.de

Abstract

Many visual representations for trees have been developed in information and software visualization. One of them are cone trees, a well-known three-dimensional representation for trees. This paper is based on an approach for constructing cone trees bottom-up. For this approach, an optimal layout for these trees is given together with a proof that based on the assumptions, there can be no better layouts. This comprises special cases, an optimal constant for the general case, and a post-processing step improving the layout.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Cone Trees, Information Visualization, Tree Layout

Digital Object Identifier 10.4230/DFU.SciViz.2010.11

1 Introduction

Cone trees are a well-known three-dimensional representation for trees. Based on the original work by Robertson et al. in 1991 [4], several different methods for the computation of the layout have been proposed. While the original approach implemented a top-down approach, in [1] a bottom-up approach was chosen to layout the cone tree. Unfortunately, some details of the algorithm were not published. A compensation factor for the computation was motivated and introduced, but its value or computation has not been given explicitely.

In this paper, we build upon this work. We show how special cases can be computed and prove a tight bound for the compensation factor of the general case.

The paper is structured as follows. In Section 2, we review the ideas of [4], [1], and other related work. In Section 3, the general setting and the assumptions for the layout are described. In Section 4, the special cases are introduced, while in Section 5 the general case is described. There, a tight bound for the compensation factor needed is proven. Further, an optimization step is introduced in this Section, too. Finally, in Section 6, open problems are given before we conclude this paper.

2 Related Work

This work is based on [4], [1], and [3]. Cone trees were first introduced in [4] (see Section 2.1). Based on this work, a different approach for constructing cone trees was proposed in [1] (see Section 2.2). In [3], reconfigurable disk trees were proposed as an enhancement to cone trees (see Section 2.3).

Scientific Visualization: Advanced Concepts.
Editor: Hans Hagen; pp. 11–29

Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Center for Informatics (Germany)

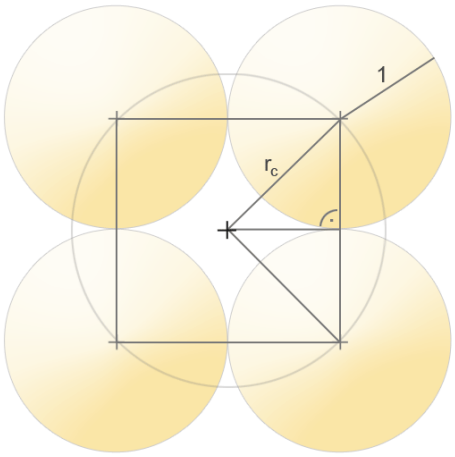

Figure 1 Counterexample.

2.1 Cone Trees

Cone trees were first presented by Robertson et al. in 1991 [4]. They are very well suited to layout trees with a large number of children. Based on the available screen size, the layout is produced in a top-down manner. Details are given in Section 3.

An evaluation of cone trees was performed by Cockburn and McKenzie in 2000 [2]. They found that for certain tasks, the participants were significantly slower than when using a 2D tree interface. Possible reasons for the bad performance of cone trees mentioned in their study comprise less familiarity with cone trees and a comparatively crude 3D interface.

2.2 Beyond Cone Trees

In [1], a different approach was chosen to layout the cone tree. The computation of the respective cones is done bottom-up. Our approach is based on this work. The details are given in Sections 3 and 5.1.

2.3 Reconfigurable Disk Trees

In [3], so-called RDTs (Reconfigurable Disk Trees) are presented. They define a reference point and an apex point together with apex height, reference height, and reference length. These additional parameters allow different cone shapes and layouts. Additionally, they provide an evaluation of the node density.

In the implementation, explicit values are given for the compensation factor introduced in [1]. Unfortunately, there are special cases where this computation will result in overlaps. One of these cases is depicted in Figure 1.

Consider four children with the radius 1. Moving them together as closely as possible leads to a square with edge length 2. Considering the triangle depicted in Figure 1, the radius is the largest edge of a triangle with a right angle and the smaller edges being 1 unit long. Thus, r_c can be computed as

$$r_c = \sqrt{1^2 + 1^2} = \sqrt{2} > 1.414$$

This is the minimal radius.

According to [3], the radius will be computed as follows:

$$\begin{aligned} inner\_sum &= 2 \cdot \sum_k out\_rad_k \\ &= 2 \cdot (1+1+1+1) \\ &= 8 \\ max\_child\_radius &= \max_k \{out\_rad_k\} \\ &= \max(1,1,1,1) \\ &= 1. \end{aligned}$$

It follows that

$$inner\_sum = 8 > 2 \cdot \pi \cdot 1 = 2 \cdot \pi \cdot max\_child\_radius.$$

Thus, the radius will be approximated as

$$rad = \frac{inner\_sum}{2 \cdot \pi} = \frac{8}{2 \cdot \pi} < 1.28.$$

Thus, $rad < r_c$, which leads to overlapping circles for this case.

3 General Setting

Cone trees are essentially built as follows. The root of the tree is placed on top. All children of a node are placed at a fixed distance below this node forming a circle. If the node is projected onto the plane of this circle, it will be projected onto the center of the circle (see also Figure 2). In Figure 3, one cone of a tree is shown. The parent node (yellow) is connected to its children using a transparent cone.

The approach chosen in [4] takes a fixed 3D space and fits the cone tree into this space. The height is divided by the depth of the tree giving the height of each cone. The radius of each circle is progressively reduced from top to bottom. The disadvantage of this approach is that the size of the children will have to be adapted to the size of the circle. For large trees, the circles containing the leaf nodes will be either very small or will not be visible at all.

Another approach was taken in [1]. There, a bottom-up approach was chosen. Each leaf element of the tree has a bounding circle. This circle gives the size of the leaf element. All leaf elements are arranged in a circle such that they do not overlap. The bounding circle of all elements arranged in the bottom circle of the cone gives the size of the cone. The whole cone, leaves plus parent node, is placed in the bottom circle of the next cone one level higher.

Here, we describe the adaptation of the ideas presented in [1]. We describe the construction of one cone. In Figure 2, a 2D view of the circle at the base of the cone is given. The radii of the bounding circles of the children determine the radius of the circle of this cone. In Figure 3, a 3D view is given. The parent node is placed at the top of the cone. All cone elements and variables are described next.

Given are a parent node N_p of the cone and its n child nodes $N_1 \dots N_n$. The size of each node is given by its radius $r_p, r_1, \dots, r_n$. The cone is described by the radius r_c of the circle at its base and its height h_c. Further, the radius r_b of the bounding circle of the children is needed. In the Table 1, these entities are summarized.

Currently, the height of the cone is always set to a predefined constant value $h_c = const$.

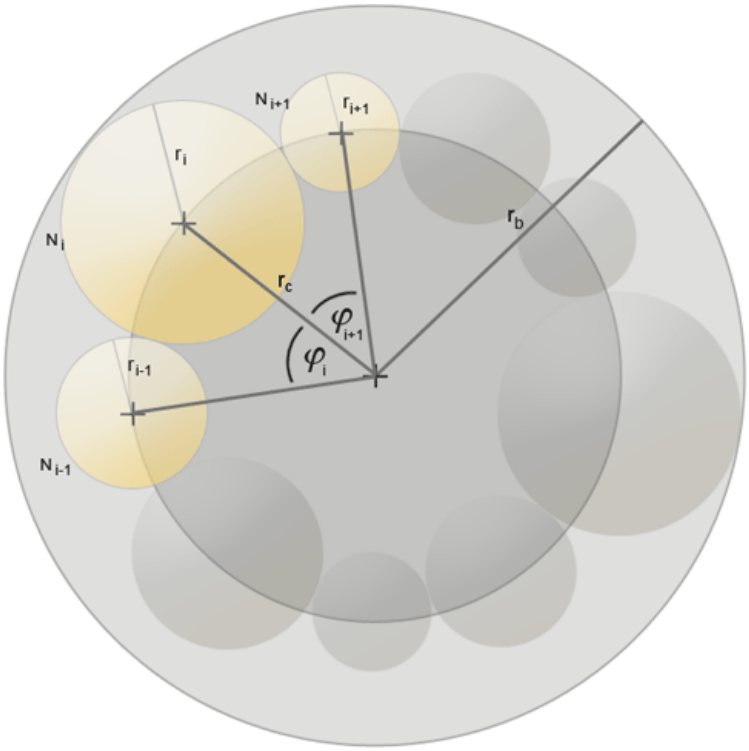

Figure 2 Construction of one cone element: 2D view.

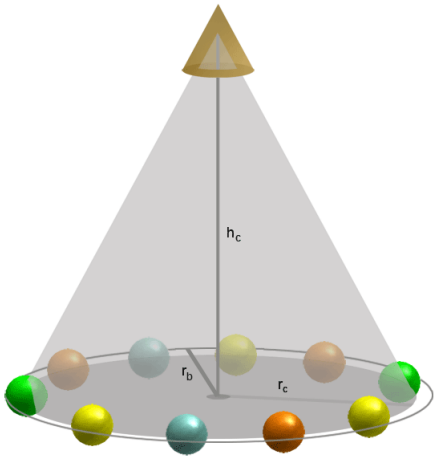

Figure 3 Construction of one cone element: 3D view.

Table 1 Abbreviations.

Abbreviation	Element
N	Node
S	Shape representing a node
R	Radius
H	Height
φ	Angle between two child nodes
C	Circumference
p subscript	Parent node
$1 \dots n$ subscript	Child node number $1 \dots n$
c subscript	Cone
b subscript	Boundary
G subscript	Glyph representing a node

4 Special Cases

4.1 No Children

If there are no children, then the radius and the bounding radius of the cone are set to the bounding radius of the glyph r_G representing the parent node N_p of the cone:

$$r_c = r_b = r_G.$$

4.2 One Child Node

If the number of child nodes is $n = 1$, then the child node is positioned directly below the parent node. Placing a child at a certain position either means placing a leaf at this position or placing the parent node of the cone representing the subtree at this position. See Figure 4 for a depiction of this situation.

The bounding radius of the cone is set to the maximum of the radius of the cone and the bounding radius r_G of the glyph representing the parent node N_p of the cone:

$$r_b = \max(r_G, r_c).$$

4.3 Two Child Nodes

If the number of children is $n = 2$, then an optimal radius and an optimal bounding radius can be computed (see Figure 5).

The cone can be computed as

$$\begin{aligned} r_c &= \frac{r_1 + r_2}{2} \\ r_b &= r_c + \max(r_1, r_2) \\ \varphi_1 &= 0^\circ \\ \varphi_2 &= 180^\circ. \end{aligned}$$

The position of the first child is at an angle of $\varphi_1 = 0^\circ$ and at a distance of r_c from the center of the circle. The position of the second child is at an angle of $\varphi_2 = 180^\circ$ and at a distance of r_c from the center of the circle.

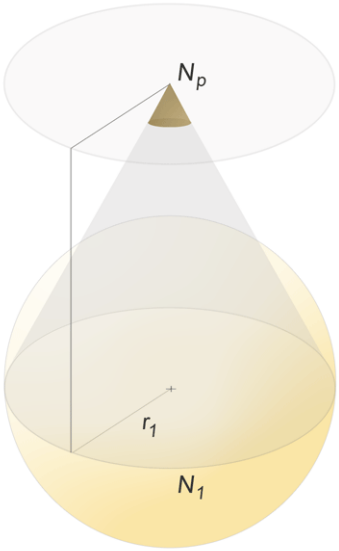

Figure 4 One child node.

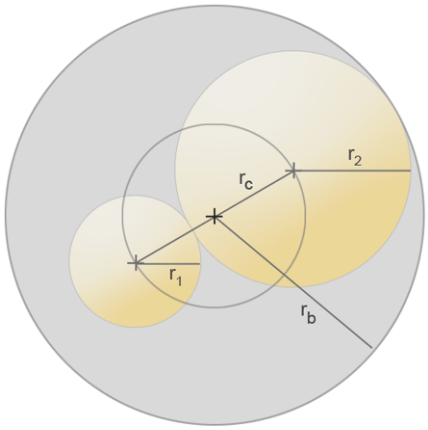

Figure 5 Two child nodes

Remark: If the radii are different, then the bounding circle is not the smallest bounding circle containing both children. Its radius is

$$\frac{|r_1 - r_2|}{2}$$

larger. One possible optimization would be to use the radius

$$r'_b = r_1 + r_2.$$

Then, the parent node has to be put at the position between the two child nodes that is the center of the bounding circle. But then the cone would no longer be a right circular cone but an oblique cone.

4.4 Three Child Nodes

If the number of children $n = 3$, then an optimal radius and an optimal bounding radius can be computed, too. Let r_1, r_2, r_3 be the radii of the children, such that without loss of generality $r_1 \geq r_2 \geq r_3$. Let A, B, C be the corners of a triangle with the edges of that triangle being

$$\begin{aligned} a &= r_1 + r_2 \\ b &= r_1 + r_3 \\ c &= r_2 + r_3. \end{aligned}$$

That is, A is the center of the child having radius r_3, B is the center of the child having radius r_2, and C is the center of the child having radius r_1.

A triangle is acute if it satisfies the following set of inequalities:

$$\begin{aligned} a^2 + b^2 &> c^2 \\ b^2 + c^2 &> a^2 \\ c^2 + a^2 &> b^2. \end{aligned}$$

If the triangle is acute, then the circumcenter of the triangle lies inside the triangle. Otherwise, it lies outside. If the circumcenter lies inside the triangle, then the children can be positioned on the circumcircle (see Figure 6). In this case, the order of the radii is not important. Let A denote the first child N_1, B denote the second child N_2, and C denote the third child N_3. The following equations can be used to compute the radius of the circumcircle and the angles of the child positions:

$$\begin{aligned} \beta &= \arccos \frac{a^2 + c^2 - b^2}{2 \cdot a \cdot c} \\ r_c &= \frac{b}{2 \cdot \sin \beta} \\ \varphi_1 &= 0^\circ \\ \varphi_2 &= \arccos \frac{r_c^2 + r_c^2 - b^2}{2 \cdot r_c \cdot r_c} \\ &= \arccos \frac{2 \cdot r_c^2 - b^2}{2 \cdot r_c^2} \\ \varphi_3 &= \arccos \frac{r_c^2 + r_c^2 - c^2}{2 \cdot r_c \cdot r_c} \\ &= \arccos \frac{2 \cdot r_c^2 - c^2}{2 \cdot r_c^2}. \end{aligned}$$

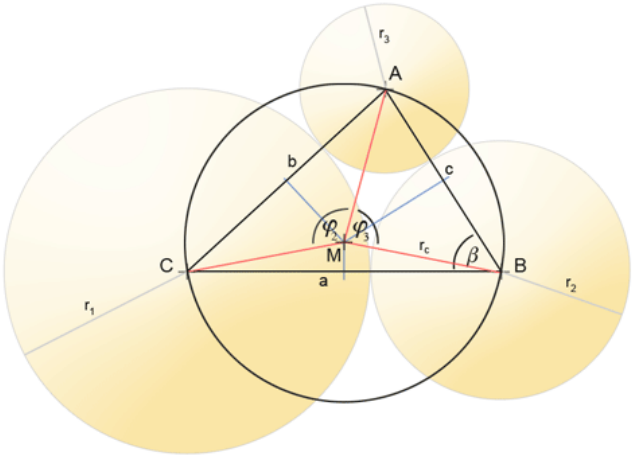

Figure 6 Three child nodes forming an acute triangle: Computation.

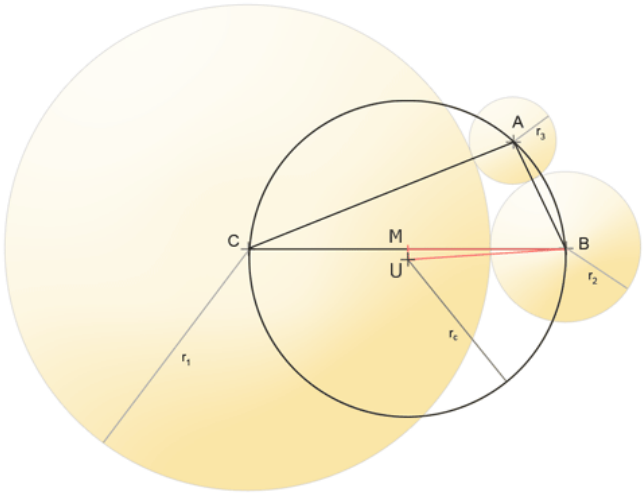

Figure 7 Three child nodes forming a non-acute triangle: Moving the circumcenter on the triangle edge.

If the circumcenter of the triangle lies outside the triangle, we get the situation depicted in Figure 7.

Consider the triangle $\triangle UMB$ where U denotes the circumcenter of the triangle and M the midpoint of the edge $\overline{BC}$. As U lies on the perpendicular bisector of $\overline{BC}$, the triangle has a right angle at the corner M. Therefore, $|\overline{MB}| < |\overline{UB}|$. Thus, using M as center and $|\overline{MB}|$ as the radius of the cone's circle results in a smaller circle. This smallest circle containing all

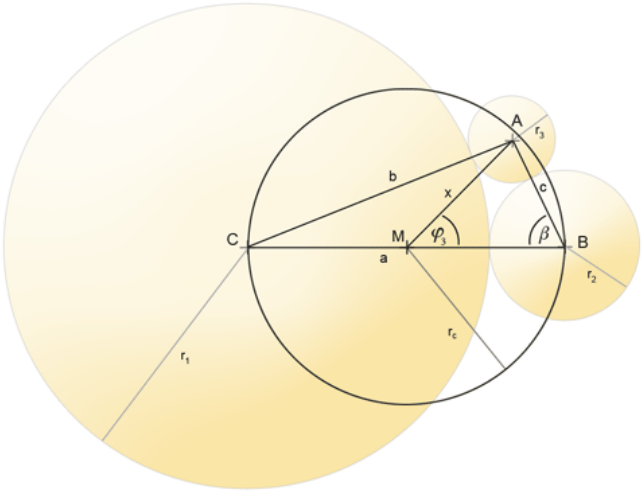

Figure 8 Three child nodes forming a non-acute triangle: Computation.

children can be constructed as follows (see also Figure 8):

$$\begin{aligned} r_c &= \frac{a}{2} \\ r_b &= r_c + r_1 \\ \varphi_1 &= 0° \\ \varphi_2 &= 180° \\ \varphi_3 &= \arccos \frac{x^2 + r_c^2 - c^2}{2 \cdot x \cdot r_c} \end{aligned}$$

with

$$\begin{aligned} \cos\beta &= \frac{a^2 + c^2 - b^2}{2 \cdot a \cdot c} \\ x^2 &= c^2 + r_c^2 - 2 \cdot c \cdot r_c \cdot \cos\beta \\ &= c^2 + r_c^2 - \frac{r_c \cdot (a^2 + c^2 - b^2)}{a}. \end{aligned}$$

In this case, the smallest child with its center A will be moved away from M along a line through M and A (see Figure 9). With respect to the closeness of the children, the circumcircle would be the better solution. But then the radius would be larger.

4.5 Equally Sized Child Nodes

Another special case that can be easily computed is all children having equal size, that is

$$\forall i : r_i = \overline{r}.$$

Then, the children can be placed at the vertices of an equilateral polygon (see Figure 10).

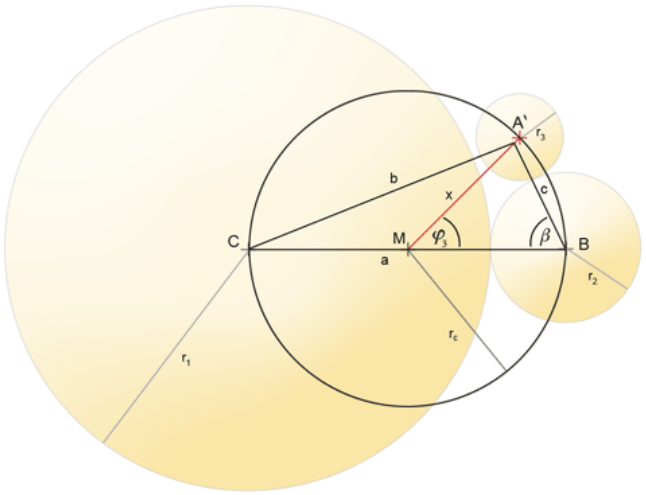

Figure 9 Moving A onto the circle.

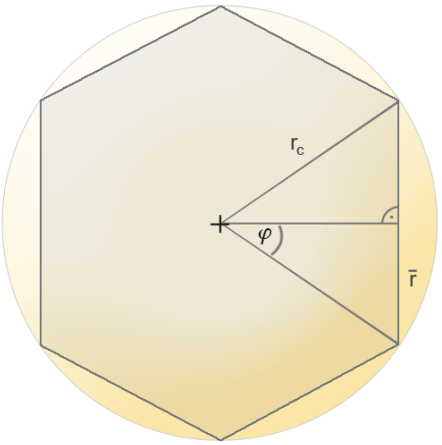

Figure 10 Circumcircle of an equilateral polygon.

The radius of the cone can be computed directly using the following equations:

$$\begin{aligned} \frac{\overline{r}}{\sin\varphi} &= \frac{r_c}{\sin 90^\circ} \\ \varphi &= \frac{2\cdot\pi}{2\cdot n} \\ &= \frac{\pi}{n} \\ &\Downarrow \\ r_c &= \frac{\overline{r}}{\sin\frac{\pi}{n}}. \end{aligned}$$

Remark: A comparison with the approximation given in Section 5.2 yields:

$$\begin{aligned} r'_c &= \frac{2 \cdot \sum_{i=1}^{n} r_i}{2 \cdot \pi} \cdot \frac{\pi}{2} = \frac{n \cdot \overline{r}}{2} \\ r''_c &= \frac{\overline{r}}{\sin \frac{\pi}{n}} \\ &\Downarrow \\ \frac{r'_c}{r''_c} &= \frac{\frac{n \cdot \overline{r}}{2}}{\frac{\overline{r}}{\sin \frac{\pi}{2}}} = \frac{n \cdot \overline{r} \cdot \sin \frac{\pi}{2}}{2 \cdot \overline{r}} = \frac{n \cdot \sin \frac{\pi}{2}}{2} \\ &\Downarrow \\ \lim_{n \to \infty} \frac{r'_c}{r''_c} &= \lim_{n \to \infty} \frac{n \cdot \sin \frac{\pi}{2}}{2} = \frac{\pi}{2}. \end{aligned}$$

This corresponds to the approximation of a circle through an equilateral polygon. Finally, in the limit the circumference of the polygon is equal to the circumference of the circle and thus the error is equal to the compensation factor introduced. On the other hand, for $n = 2$, the quotient is equal to 1 and thus the compensation factor is needed.

5 General Case

The general case allows computing only an approximation of the cone. First, the construction of the circle is given in Section 5.1. In order to compute the radius, a compensation factor is used. A tight bound of this compensation factor is motivated and proven in Section 5.2. Finally, it is possible to improve the construction with a post-processing step. This optimization is presented in Section 5.3.

5.1 Introduction

In case a parent node has more than three children, $n > 3$, an exact computation of the inner circle is no longer possible, except for special cases. One special case would be that all children have the same bounding radius (see Section 4.5). In general, the cone and the positions of the children can be computed as follows (see also Figure 11 and [1]).

The circumference of the cone's base circle can be approximated as

$$\widetilde{c_c} \widetilde{=} 2 \cdot \sum_{i=1}^{n} r_i.$$

An approximation of the radius of the cone's base circle can then be computed as

$$\widetilde{r_c} = \frac{\widetilde{c_c}}{2 \cdot \pi}.$$

The radius used for the cone's base circle is obtained by multiplying the approximated radius by a compensation factor f

$$r_c = \widetilde{r_c} \cdot f.$$

If the number of children is small or if there is a large difference between the radii of the smallest and the largest child, then the circumference will be underestimated. The compensation factor f was motivated and introduced in [1], but no formula for its computation has been given. The special cases for less than four children, $n < 4$, have already been

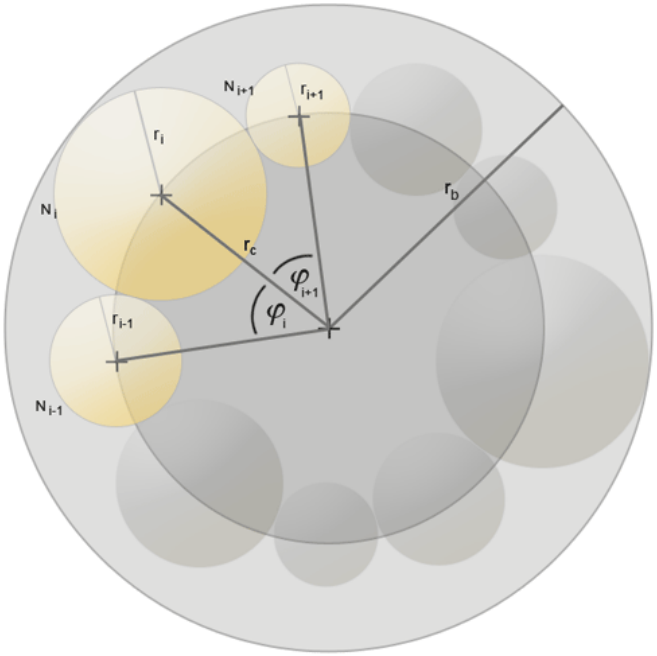

Figure 11 Cone with more than three children.

addressed in Section 4. Here, the maximal error between approximated and minimal circumference needed is computed. It is given by the following equation (see Section 5.2):

$$\epsilon = \frac{c_c}{\widetilde{c_c}} \leq \frac{\pi}{2}.$$

Therefore, choosing

$$f = \frac{\pi}{2}.$$

as compensation factor is sufficient.

The radius of the bounding circle is computed using the following equation:

$$r_b = r_c + \max_{i=1\ldots n} \{r_i\}.$$

Each child is positioned at distance r_c from the center of the cone's circle. The first child is always positioned at $\varphi_1 = 0°$. The angle φ_i between two children N_i and N_{i-1}, $i > 1$ is computed as

$$\varphi_i = \frac{r_{i-1} + r_i}{r_c}.$$

Hence, the children N_i, $i > 1$ are positioned at

$$\overline{\varphi_i} = \sum_{j=1}^{i} \varphi_j.$$

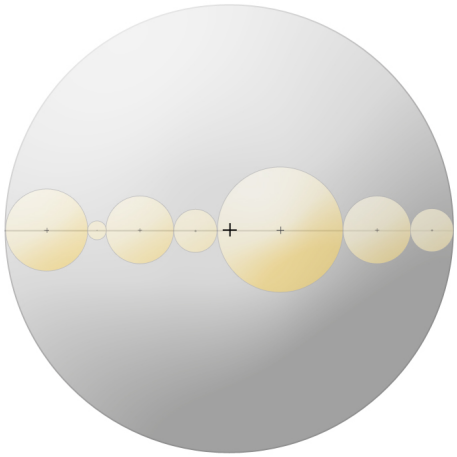

Figure 12 Identification of π as bound of the compensation factor; line-up of the children.

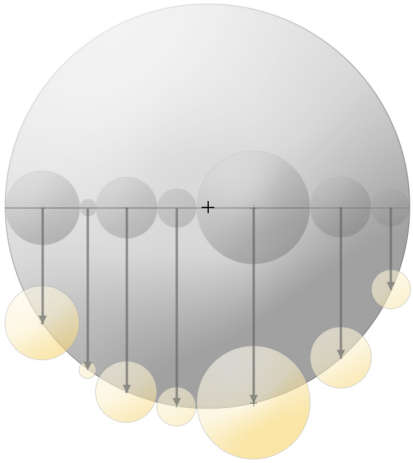

Figure 13 Explanation of π as bound of the compensation factor; moving children onto the boundary.

5.2 Computation of the Compensation Factor

The compensation factor needed in the previous section can be estimated as follows. First of all, we consider the situation depicted in Figure 12. All children are lined up yielding the diameter d of a circle

$$d = 2 \cdot \sum_{i=1}^{n} r_i.$$

Now, the children can be moved perpendicular to the line from the line onto the boundary of the circle (see Figure 13). That is, all children can be placed on a circle having radius r_c

and circumference c_c, such that

$$\begin{aligned} r_c &= \frac{d}{2} = \sum_{i=1}^{n} r_i \\ c_c &= 2 \cdot \pi \cdot r_c = 2 \cdot \pi \cdot \sum_{i=1}^{n} r_i. \end{aligned}$$

The quotient between the chosen circumference c_c and the estimated circumference $\widetilde{c_c}$ is

$$\frac{c_c}{\widetilde{c_c}} = \frac{2 \cdot \pi \cdot \sum_{i=1}^{n} r_i}{2 \cdot \sum_{i=1}^{n} r_i} = \pi.$$

As can be seen from Figure 13, only one half of the circle is used for placing the children. Thus, the compensation factor is much too large. The conjecture is that using

$$f = \frac{\pi}{2}$$

as compensation factor would be sufficient.

Consider the following inequalities:

$$\begin{aligned} \frac{c_c}{\widetilde{c_c}} \leq \frac{\pi}{2} \quad &\Leftrightarrow \quad \frac{2 \cdot \pi \cdot r_c}{2 \cdot \sum_{i=1}^{n} r_i} \leq \frac{\pi}{2} \\ &\Leftrightarrow \quad 2 \cdot \pi \cdot r_c \leq \frac{\pi}{2} \cdot 2 \cdot \sum_{i=1}^{n} r_i \\ &\Leftrightarrow \quad 4 \cdot r_c \leq 2 \cdot \sum_{i=1}^{n} r_i. \end{aligned}$$

In order to show the latter relation, consider the following situations. All children are arranged such that the centers of their bounding circles form a convex polygon. The distance between two children is the sum of their radii. Compute a minimal bounding circle around this convex polygon. Please note that this construction can always be performed. Then, there are two possible situations:

1. Exactly two vertices of the convex polygon are lying on the border of the minimal bounding circle (see Figure 14).
2. Three or more vertices of the convex polygon are lying on the border of the minimal bounding circle (see Figure 15).

If no or only one vertex lies on the border of the circle, then the circle would not be minimal.

Considering the first case, if exactly two vertices of the convex polygon are lying on the border of the bounding circle, then the distance between these vertices is equal to the diameter of the circle. Otherwise, the bounding circle would not be minimal.

Let A and B be the vertices lying on the minimal bounding circle. Then, we get

$$r_c = \frac{|\overline{AB}|}{2}$$

and the above equation is certainly fulfilled:

$$4 \cdot r_c = 4 \cdot \frac{\overline{|AB|}}{2} = 2 \cdot \overline{|AB|} \leq 2 \cdot \sum_{i=1}^{n} r_i.$$

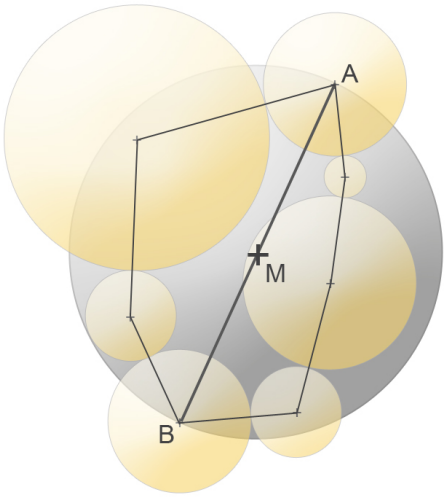

Figure 14 Explanation of the compensation factor: first case.

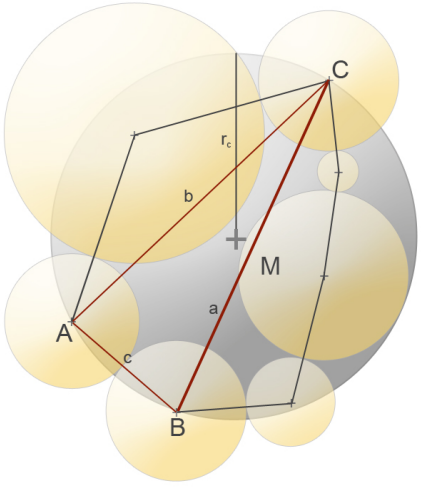

Figure 15 Explanation of the compensation factor: second case.

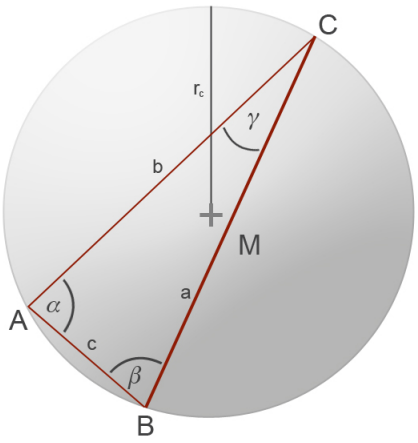

Figure 16 Explanation of the compensation factor: second case.

This holds because the path from A to B using lines of the polygon is certainly longer than the direct path between A and B (triangle inequality).

If three or more vertices are lying on the boundary of the minimal bounding circle, then the result is obtained as follows. Consider any triangle with all vertices lying on the boundary. Then, the bounding circle is also the circumcircle of the triangle. There is at least one triangle with the circumcenter lying inside the triangle or on one of its sides. Otherwise, for each triangle, the circumcenter lies outside the triangle. This is the second situation of the special case described in Section 4.4 and the circle would not be minimal.

Now, choose a triangle such that the circumcenter lies inside the triangle or on one of its sides. If the center of the circumcircle lies on one of the sides of the triangle, then this side is the diameter of the circle and we have the same situation as in the first case. Otherwise, let a, b, c be the sides of the triangle and A, B, C its vertices.

If

$$4 \cdot r_c \leq |a| + |b| + |c|$$

then

$$4 \cdot r_c \leq |a| + |b| + |c| \leq 2 \cdot \sum_{i=1}^{n} r_i.$$

The last inequality holds because a, b, and c are the shortest connections between A, B, and C. Every path on the polygon is longer (triangle inequality). Thus, it is sufficient to show that the first inequality holds.

Consider Figure 16. First of all, the law of sines implies the following equations:

$$\begin{aligned} |a| + |b| + |c| &= 2 \cdot r_c \cdot \sin\alpha + 2 \cdot r_c \cdot \sin\beta + 2 \cdot r_c \cdot \sin\gamma \\ &= 2 \cdot r_c \cdot (\sin\alpha + \sin\beta + \sin\gamma). \end{aligned}$$

Thus, it is sufficient to show

$$\sin\alpha + \sin\beta + \sin\gamma \geq 2$$

because then

$$\begin{aligned} |a| + |b| + |c| &= 2 \cdot r_c \cdot (\sin\alpha + \sin\beta + \sin\gamma) \\ &\geq 2 \cdot r_c \cdot 2 \\ &= 4 \cdot r_c. \end{aligned}$$

In order to show that the sum of the sine of the angles is greater or equal to two, consider the following conditions. As the triangle is acute, we have

$$A, B, C \leq 90°.$$

Therefore, we get

$$A + B \geq 90° \Rightarrow A \geq 90° - B.$$

From this follows

$$\begin{aligned} \sin\alpha &\geq \cos\beta \\ \sin\beta &\geq \cos\alpha. \end{aligned}$$

Using these, we get

$$\begin{aligned} \sin\alpha + \sin\beta + \sin\gamma &= \sin\alpha + \sin\beta + \sin(180° - (\alpha + \beta)) \\ &= \sin\alpha + \sin\beta + \sin(\alpha + \beta) \\ &= \sin\alpha + \sin\beta + \sin\alpha \cdot \cos\beta + \cos\alpha \cdot \sin\beta \\ &\geq \sin\alpha + \sin\beta + \cos^2\beta + \cos^2\alpha \\ &= \sin\alpha + \sin\beta + 1 - \sin^2\beta + 1 - \sin^2\alpha \\ &= 2 + \sin\alpha \cdot (1 - \sin\alpha) + \sin\beta \cdot (1 - \sin\beta) \\ &\geq 2. \end{aligned}$$

From this follows the claim about the compensation factor at the beginning of this section.

5.3 Optimization Step

Having computed the layout as described in Section 5.1, an optimization step can be performed (see Figure 17). This is done by computing for each child an optimization factor

$$\begin{aligned} f_i &= \frac{r_i + r_{i+1}}{d(r_i, r_{i+1})} \\ &= \frac{r_i + r_{i+1}}{\sqrt{r_c \cdot r_c + r_c \cdot r_c - 2 \cdot r_c \cdot r_c \cdot \cos\varphi}} \\ &= \frac{r_i + r_{i+1}}{\sqrt{(2 - 2 \cdot \cos\varphi)} \cdot r_c^2}. \end{aligned}$$

where $d(r_i, r_{i+1})$ denotes the distance between r_i and r_{i+1}. Then, an optimized radius can be computed and used for the layout of the children:

$$\hat{r}_c = \max(f_i) \cdot r_c.$$

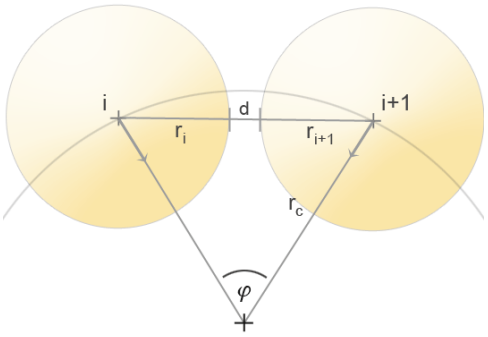

Figure 17 Optimization step.

6 Future research and open problems

Although the layout computed is optimal under the assumptions presented in Section 3 and Section 5.1, there is still space for improvements.

First of all, the proof for the general case is quite long. The question is whether there is a more elegant proof of the bound for this case.

The second question is whether there is an easy way to extend the special cases. Further, only a small number of special cases have been considered, namely, zero to three children and equally sized children. There might be further special cases, leading to simple and efficient solutions.

While this paper focused on the computation of the circle parameters, another parameter is the height of the cone. The question is whether there is an elegant way to compute an optimal height.

Finally, the implications of this work on RDTs can be researched, too.

7 Conclusion

Performing a bottom-up construction of cone trees, formulas for the optimal computation of the cone's base circles have been proposed. This includes special cases for zero to three children and for equally sized children as well as the general case. For the latter, an optimal compensation factor has been motivated and proven. Further, an optimization step has been introduced for this case. This allows implementing cone trees bottom-up in an optimal way.

8 Acknowledgment

Many thanks go to Timo Klein who provided the illustrations of this paper. The authors also wish to thank the participants of the Dagstuhl Seminar 07221 "Information Visualization - Human-Centered Issues in Visual Representation, Interaction, and Evaluation" for valuable comments on a presentation of a preliminary version of this paper. Further, we thank the unknown referees of TVCG of valuable comments on this paper.

References

1 Jeromy Carriere and Rick Kazman. Interacting with huge hierarchies: Beyond cone trees. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'95)*, pages 74–81, Atlanta, 1995.

2 Andy Cockburn and Bruce McKenzie. An evaluation of cone trees. In *People and Computers XV: Proceedings of the British Computer Society Conference on Human Computer Interaction 2000*, pages 425–436. Springer Verlag, 2000.

3 Chang-Sung Jeong and Alex Pang. Reconfigurable disc trees for visualizing large hierarchical information space. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'98)*, pages 19–25, North Carolina, 1998.

4 George G. Robertson, Jock D. Mackinlay, and Stuart Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proceedings of ACM CHI'91*, pages 189–194, New Orleans, USA, 1991.