On Balloon Drawings of Rooted Trees

Chun-Cheng Lin and Hsu-Chun Yen*

Dept. of Electrical Engineering, National Taiwan University,
Taipei, Taiwan 106, ROC
sanlin@cobra.ee.ntu.edu.tw, yen@cc.ee.ntu.edu.tw

Abstract. Among various styles of tree drawing, *balloon drawing*, where each subtree is enclosed in a circle, enjoys a desirable feature of displaying tree structures in a rather balanced fashion. We first design an efficient algorithm to optimize angular resolution and aspect ratio for the balloon drawing of rooted unordered trees. For the case of ordered trees for which the center of the enclosing circle of a subtree need not coincide with the root of the subtree, flipping the drawing of a subtree (along the axis from the parent to the root of the subtree) might change both the aspect ratio and the angular resolution of the drawing. We show that optimizing the angular resolution as well as the aspect ratio with respect to this type of rooted ordered trees is reducible to the perfect matching problem for bipartite graphs, which is solvable in polynomial time. Aside from studying balloon drawing from an algorithmic viewpoint, we also propose a local magnetic spring model for producing dynamic balloon drawings with applications to the drawings of galaxy systems, H-trees, and sparse graphs, which are of practical interest.

1 Introduction

Since the majority of algorithms for drawing rooted trees take linear time, rooted tree structures are suited to be used in an environment in which real-time interactions with users are frequent. Among existing algorithms in the literature for drawing rooted trees, triangular tree drawing [8], radial or hyperbolic drawing [5], and balloon drawing [1, 3, 6] with respect to cone trees [9] are popular for visualizing hierarchical graphs. Our concern in this paper is a *balloon drawing* of a rooted tree which is a drawing having the following properties: (1) all the children under the same parent are placed on the circumference of the circle centered at their parent, (2) there exist no edge crossings in the drawing, and (3) with respect to the root, the deeper an edge is, the shorter its drawing length becomes.

Each subtree in the balloon drawing of a tree is enclosed entirely in a circle, which resides in a *wedge* whose end-point is the parent node of the subtree. The radius of each circle is proportional to the number of descendants associated with the root node of the subtree. The ray from the parent to the root of the subtree divides the wedge into two sub-wedges. Depending on whether the two sub-wedge angles are required to be identical or not, a balloon drawing can

* Corresponding author. Supported in part by NSC Grant 94-2213-E-002-086, Taiwan.

P. Healy and N.S. Nikolov (Eds.): GD 2005, LNCS 3843, pp. 285–296, 2005.

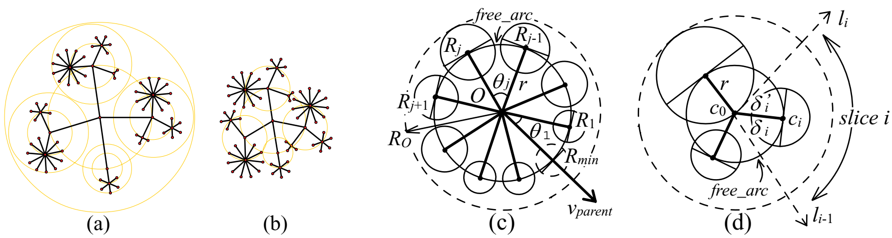

Fig. 1. (a) and (b) illustrate balloon drawings. (c) and (d) illustrate the SNS model. Note that in (c), node O is not the root, and the edge between O and its parent goes through a circle with radius R_{min}; (d) is a star graph centered at c_0.

further be divided into two types: drawings with *even angles* (see Figure 1(a)) and drawing with *uneven angles* (see Figure 1(b)).

The main aesthetic criteria on balloon drawing are *angular resolution* and *aspect ratio. Angular resolution* refers to the smallest angle between two adjacent edges incident to the common node in straight-line drawing, whereas *aspect ratio* is defined as the ratio of the largest angle to the smallest angle formed by two adjacent edges incident to the common node in straight-line drawing. A tree layout with a small aspect ratio often enjoys a very balanced view of a tree.

It is not hard to observe that with respect to a rooted *unordered* tree, changing the order in which the children of a node are listed affects the angular resolution as well as the aspect ratio of the drawing. Hence an interesting question arises: *How to find an embedding of a rooted unordered tree such that the balloon drawing of the tree (of the even angle type) has the maximum angular resolution and the minimum aspect ratio?* In the first part of this paper, we demonstrate an efficient algorithm which is guaranteed to yield an optimal balloon drawing in terms of (maximum) angular resolution and (minimum) aspect ratio.

Now consider the case of the uneven angle type. Allowing uneven angles introduces another dimension of flexibility as far as optimizing angular resolution and aspect ratio is concerned. Even if the embedding (ordering) of a tree is given, flipping the drawing of a subtree along the axis going through the parent and the root of the subtree might change the angular resolution and the aspect ratio of the drawing. Notice in the uneven angle case, the angles on the two sides of the axis might not be equal. A related question is: *How to flip uneven angles in the balloon drawing of a rooted ordered tree to achieve optimality in angular resolution and aspect ratio?* Notice in the above, the embedding of the underlying tree is fixed. As it turns out, we are able to reduce the above problem to that of *perfect matching* of bipartite graphs, which admits a polynomial time solution.

Aside from the above two algorithmic issues related to balloon drawing, the second part of this paper deals with the design and implementation of a *force-directed method* (see, e.g., [2, 11]) to provide *dynamic* balloon drawings. Scalability, interactivability, and predictability make dynamic drawing interesting and important in the issues arising from many applications of information visualization. More details about our approach will be given in Section 5.

2 Preliminaries

A tree is a connected acyclic graph. A rooted tree where the order of the subtrees is significant and fixed is called an *ordered tree*; otherwise, it's called *unordered*. A *star graph* with $n+1$ nodes is a rooted tree in which the root node is of degree n and the others are of degree one. Given a drawing of graph G, the *angular resolution* at node v refers to the smallest angle formed by two adjacent edges incident to the common node v in the drawing of G. The *angular resolution* of a drawing of G is defined as the minimum angular resolution among all nodes in G. The *aspect ratio* of a drawing of G is the ratio of the largest angular resolution to the smallest angular resolution in the drawing. The angular resolution (resp., aspect ratio) of a graph is in the range of $(0°, 360°)$ (resp., $[1, \infty)$).

There exist two models in the literature for generating *balloon drawings* of trees. Given a node v, let $r(v)$ be the radius of the drawing circle centered at v. If we require that $r(v) = r(w)$ for arbitrary two nodes v and w that are of the same depth from the root of the tree, then such a drawing is called a balloon drawing under the *fractal model* [4]. Under this model, if r_m and r_{m-1} are the lengths of edges at depths m and $m-1$, respectively, then

$$r_m = \gamma \times r_{m-1} \tag{1}$$

where γ is the predefined ratio ($0 < \gamma < 1$) associated with the fractal drawing.

Unlike the fractal model, the *subtree with nonuniform sizes* (abbreviated as *SNS*) model [1, 3] allows subtrees associated with the same parent to reside in circles of different sizes, and hence, the drawing based on this model often results in a clearer display on large subtrees than that under the fractal model.

Theorem 1 (see [1, 3]). *Given a rooted ordered tree T with n nodes, a balloon drawing under the SNS model can be obtained in $O(n)$ time in a bottom-up fashion with the edge length and the angle between two adjacent edges according to equations (2) and (3) respectively:*

$$r = C/(2\pi) \cong (2\sum_j R_j)/(2\pi) \tag{2}$$

$$\theta_j \cong (R_{j-1} + free\_arc + R_j)/r \tag{3}$$

(see Figure 1(c)(d)) where r is the radius of the inner circle centered at node O; C is the circumference of the inner circle; R_j is the radius of the outer circle enclosing all subtrees of the j-th child of O, and R_O is the radius of the outer circle enclosing all subtrees of O; since there exist the gap between C and the sum of all diameters in Equation (2), we can distribute to every θ_j the gap between them evenly, denoted by $free\_arc$.

Note that the trees considered in [1, 3] are ordered. Since all the angles incident to a common node are the same in the fractal model, changing the ordering of the subtrees of a node at any level does not affect the angular resolution (nor the aspect ratio). Under the SNS model, however, the ordering of subtrees is critical as far as angular resolution and aspect ratio are concerned. Our goal is to devise an algorithm for optimizing the angular resolution and aspect ratio of the balloon drawing of a rooted unordered tree (under the SNS model).

3 Balloon Drawings with Even Angles

First, consider the following way of drawing a star graph with circles of nonuniform size attached to the children of the root (see Figure 1(d) for an example).

Definition 1. *The balloon drawing of a star graph with children of nonuniform size is a drawing in which*

1. *circles associated with different children of the root do not overlap, and*
2. *all the children of the root are placed on the circumference of a circle centered at the root.*

Let S be a star graph with $n+1$ nodes $\{c_0, c_1, ..., c_n\}$, where c_0 is the root. It can easily be seen from Figure 1(d) that, in a balloon drawing of S, the circle centered at the root is divided into n *wedges* (or *slices*) each of which accommodates a circle associated with a child of c_0. Let δ_i (resp., δ'_i) be the angle between rays $\overrightarrow{l_{i-1}}$ and $\overrightarrow{c_0c_i}$ (resp., $\overrightarrow{l_i}$ and $\overrightarrow{c_0c_i}$). The balloon drawing is said to be of *even angle* if $\delta_i = \delta'_i$, for all $1 \leq i \leq n$. That is, $\overrightarrow{c_0c_i}$ divides the respective wedge into two equal sub-wedges; otherwise the drawing is said to be of *uneven angle*. In this section, we only consider balloon drawings of even angle. More will be said about the uneven angle case in Section 4.

Let $\theta_i, 1 \leq i \leq n$, be the degree of the wedge angle enclosing the circle centered at node c_i. (In Figure 1(d) $\theta_i = \delta_i + \delta'_i = 2\delta_i$, assuming the even angle case.) An ordering of the children of c_0 is simply a *permutation* σ of $\{1, ..., n\}$, which specifies the placements of nodes $c_1, ..., c_n$ (and their associated circles) along the circumference of the circle centered at c_0 in the balloon drawing. More precisely, the children are drawn in the order of $c_{\sigma_1}, c_{\sigma_2}, ..., c_{\sigma_n}$, in which c_{σ_i} and $c_{\sigma_{i\oplus 1}}$, $1 \leq i \leq n$, are neighboring nodes. [1] With respect to σ, the degree of the angle between $\overrightarrow{c_0c_i}$ and $\overrightarrow{c_0c_{i\oplus 1}}$ is $(\theta_{\sigma_i} + \theta_{\sigma_{i\oplus 1}})/2$. Hence, the angular resolution (denoted by $AngResl_\sigma$) and the aspect ratio (denoted by $AspRatio_\sigma$) are

$$AngResl_\sigma = \min_{1\leq i\leq n}\{\frac{\theta_{\sigma_i} + \theta_{\sigma_{i\oplus 1}}}{2}\}, AspRatio_\sigma = \{\frac{\max_{1\leq i\leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i\oplus 1}}}{2}\}}{\min_{1\leq i\leq n}\{\frac{\theta_{\sigma_i}+\theta_{\sigma_{i\oplus 1}}}{2}\}}\}. \quad (4)$$

Let Σ be the set of all permutations of $\{1, ..., n\}$. In what follows, we shall design an efficient algorithm to find a permutation that returns

$$optAngResl = \max_{\sigma\in\Sigma}\{AngResl_\sigma\} \texttt{ and } optAspRatio = \min_{\sigma\in\Sigma}\{AspRatio_\sigma\}.$$

The *optAngResl* is said to *involve* degrees of angles θ_{σ_i} and $\theta_{\sigma_{i\oplus 1}}$ if i is the value minimizing $AngResl_\sigma$ of Equation (4) w. r. t. the optimal permutation σ.

For notational convenience, we order the set of wedge angles $\theta_1, ..., \theta_n$ in ascending order as either

$$m_1, m_2, ..., m_{k-1}, m_k, M_k, M_{k-1}, ..., M_2, M_1 \texttt{ if } n \texttt{ is even,} \quad (5)$$

$$\texttt{or } m_1, m_2, ..., m_{k-1}, m_k, mid, M_k, M_{k-1}, ..., M_2, M_1 \texttt{ if } n \texttt{ is odd,} \quad (6)$$

for some k where m_i (resp. M_i) is the i-th minimum (resp. maximum) among all, and *mid* is the median if n is odd. We define $\alpha_{ij} = (M_i + m_j)/2, 1 \leq i, j \leq k$.

[1] $i \oplus 1$ denotes $(i \bmod n) + 1$.

Procedure 1. OPTBALLOONDRAWING

Input: a star graph S with n child nodes of nonuniform size.
Output: a balloon drawing of S optimizing angular resolution and aspect ratio.

– Sort the set of degrees of the wedge angles (accommodating the n nonuniform circles) into ascending order as mentioned in Equations (5) and (6).
– Output a drawing witnessed by the following circular permutation:
$(M_1, m_2, M_3, m_4, ..., \mu, (, mid), \nu, ..., M_4, m_3, M_2, m_1)$
where $\{\mu, \nu\} = \{M_k, m_k\}$ whose values depends on whether $n = 2k$ or $2k + 1$ and whether k is odd or even. Note that M_1 and m_1 are adjacent.

Recall from Figure 1(c) that, the drawing of the subtree rooted at node O is enclosed in a circle centered at O. By abstracting out the details of each of the subtrees associated with the children of O, the balloon drawing of the subtree at O can always be viewed as a balloon drawing of a star graph with children of nonuniform size rooted at O, regardless of the depth at which O resides. In addition, even if we alter the ordering of the children of O, the size of the outer circle bounding all the children of O remains the same; hence, the optimization of each of the subtrees at depth k does not affect the optimization of their parent at depth $k-1$. In view of the above, optimizing the angular resolution and the aspect ratio of a balloon drawing of a rooted unordered tree can be carried out in a bottom-up fashion. So, it suffices to investigate how to optimize the angular resolution and the aspect ratio of balloon drawing with respect to star graphs.

Theorem 2. *Procedure 1 achieves optimality in angular resolution as well as in aspect ratio for star graphs.*

Proof. (Sketch) In what follows, we only consider the case

$$\sigma = (M_1, m_2, M_3, m_4, ..., M_{k-1}, m_k, mid, M_k, m_{k-1}, ..., M_4, m_3, M_2, m_1)$$

i.e., $n = 2k + 1$ and k is odd, and assuming that degrees are all distinct; the remaining cases are similar (in fact, simpler).

Recall that $\alpha_{ij} = \frac{M_i + m_j}{2}$. One can easily see the following properties of σ:

Property (1). For each $i \in \{2, ..., k\}$, the angles of degree $(mid + m_k)/2$, $\alpha_{(i-1)i} = (M_{i-1} + m_i)/2$, $(M_k + mid)/2$, and $\alpha_{i(i-1)} = (M_i + m_{i-1})/2$ are included in σ;

Property (2). The minimum degree of σ must be $(mid + m_k)/2$ or $\alpha_{j(j-1)}$, while the maximum degree of σ must be $(M_k + mid)/2$ or $\alpha_{(l-1)l}$ for some $j, l \in \{2, ..., k\}$.

The reason behind Property (2) is that all the angles consecutively appearing in σ have the following ordering relationship:

$$\alpha_{12} > \alpha_{32} < \alpha_{34} > ... > \alpha_{j(j-1)} < ... < \alpha_{(l-1)l} > ... > \alpha_{k(k-1)} < (M_k + mid)/2$$
$$> (mid + m_k)/2 < \alpha_{(k-1)k} > ... > \alpha_{43} < \alpha_{23} > \alpha_{21} < (M_1 + m_1)/2 < \alpha_{12}.$$

Suppose δ is the permutation that witnesses *optAngResl*. ¿From Property (2), the minimum angular resolution of σ must be either $\alpha_{i,i-1}$, for some $2 \leq i \leq k$, or $(mid + m_k)/2$. In what follows, we only need to consider the case when the minimum angular resolution of σ is $\alpha_{i,i-1}$; the other case can be proved similarly.

Now if M_i is a neighbor of m_{i-1} in δ (the optimal permutation), then δ and σ have the same angular resolution and $optAngResl = \alpha_{i,i-1}$ because, otherwise,

– if $optAngResl < \alpha_{i,i-1}$ (which is the angular resolution of σ), then this contradicts that δ is optimal.
– if $optAngResl > \alpha_{i,i-1}$, then this's impossible because δ has an angle of degree $\alpha_{i,i-1} = (M_i + m_{i-1})/2$.

Hence, σ is optimal as well.

On the other hand, suppose x and y ($x < y$) are the two neighbors of m_{i-1} in δ and neither one is M_i, then both x and y must be greater than M_i; otherwise, the angular resolution of δ is smaller than $(M_i + m_{i-1})/2$ – contradicting δ being optimal. Also note that $optAngResl \geq (m_{i-1} + x)/2$. Now if we look at a partition of the set of wedge angles of S as follows:

$$\underbrace{m_1 < ...}_{R_A} < m_{i-1} < \underbrace{...}_{R_B} < M_i < \underbrace{... < x < ... < y < ... < M_1}_{R_C}. \tag{7}$$

R_A contains $i-2$ elements, which must be connected to at least $i-1$ elements of R_C; otherwise, the angular resolution in δ becomes less than $(M_i + m_{i-1})/2$ – a contradiction. R_C originally contains $i-1$ elements. However, x and y are the two neighbors of m_{i-1} – meaning that together with m_{i-1} they are tied together and cannot be separated. So effectively only '$i-2$' elements of R_C can fill the $i-1$ neighbors of R_A – which is not possible. We have a contradiction. What the above shows is that in the optimal permutation δ, a neighbor of m_{i-1} is M_i. Hence, the angular resolution of δ is $\leq (M_i + m_{i-1})/2$. Since δ witnesses *optAngResl*, $optAngResl = (M_i + m_{i-1})/2$, meaning that σ also produces *optAngResl*.

The above implies that *optAngResl* must be either $(mid + m_k)/2$ or $\alpha_{i(i-1)}$ for some $i \in \{2, ..., k\}$, which is always included in the circular permutation σ produced by Procedure 1. Similarly, we can prove that the minimum degree of the largest angle of any drawing must be $(M_k + mid)/2$ or $\alpha_{(j-1)j}$ for some $j \in \{2, ..., k\}$, which is also always in σ. Since σ simultaneously possesses both the maximum degree of the smallest angle and the minimum degree of the largest angle of any drawing, σ also witnesses the optimum aspect ratio. □

Using Procedure 1, the drawing of a rooted unordered tree which achieves optimality in angular resolution and aspect ratio can be constructed efficiently in a bottom-up fashion.

Finally, in order to give a sense perception on the advantages of our algorithm, we implement the algorithm and give a simple experimental result shown in Figure 2. The drawing in (a) based on the fractal model displays that the degrees of angles spanned by adjacent edges are identical, and hence has the best angular resolution and aspect ratio. The major drawback of fractal drawing, as seen

tree information			Angular resolution			Aspect ratio		
node num	max degree	depth	fractal	SNS	our method	fractal	SNS	our method
1000	20	4	18°	3.50°	6.33°	1	13.4961	4.6718

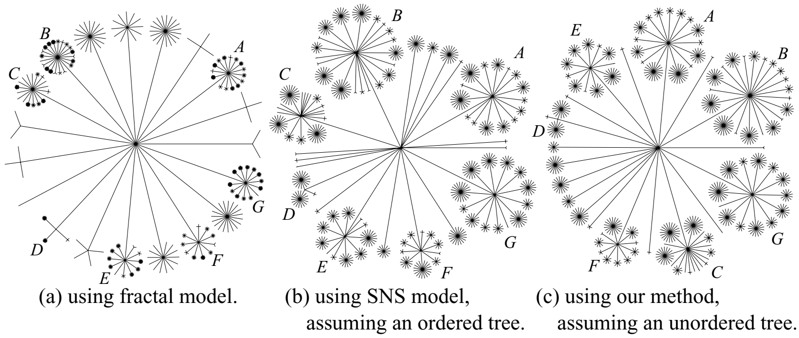

Fig. 2. An experimental result. (The above is its statistics.)

in (a), is that visibility deteriorates considerably as we move towards deeper subtrees of complicated structures. For observing more complicated subtrees clearer and easier, one may apply the SNS model (with respect to an ordered tree) to yield (b). (Note that (b) has the same ordering as that in(a).) To a certain extent, the SNS model sacrifices the angular resolution and the aspect ratio in order to gain better visibility for displaying complicated subtree structures. If the ordering of subtrees is allowed to be altered, our optimization algorithm has the ability to optimize the angular resolution and the aspect ratio under the SNS model on the balloon drawing of the rooted unordered tree, as shown in (c).

4 Balloon Drawings with Uneven Angles

The area of a balloon drawing can be measured by the size of the circle enclosing the drawing. Minimizing the area of a drawing is an important issue because any drawing needs to be rendered on a limited region. A careful examination of the approach investigated in Section 3 suggests that the area of balloon drawing generated by the SNS model may not be minimal. Part of the reason is the involvement of the so-called $free\_arc$ described in Theorem 1 and Figure 1(d), serving for the purpose of separating the enclosing circles of two neighboring subtrees. A more subtle point regarding the 'waste' of drawing space is illustrated in Figure 3, in which (a) shows the drawing of a tree under the SNS model. Let T_v be the subtree rooted at v. Based on the approach discussed in Section 3, T_v resides in a circle centered at v and the circle included in a wedge in which the ray from O to v cuts the wedge into two sub-wedges of identical size (i.e., $\theta_1 = \theta_2$). By limiting the drawing to the area formed by two lines (see t_1 and t_2 in (a)) tangent to the outer circle of children of v, the drawing area is reduced, i.e., the new wedge (in which the drawing of T_v resides) is now spanned by lines t_1 and t_2 with the degree of the wedge angle equals $\theta_3 + \theta_4$. Furthermore, the ray $\overrightarrow{Ov}$ cuts this new wedge into two possibly uneven parts (i.e., θ_3 need not be equal

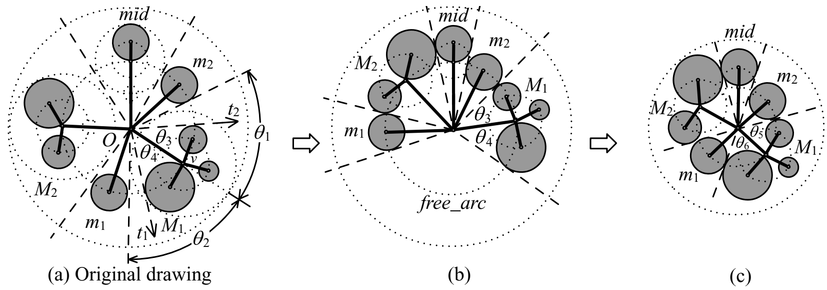

Fig. 3. Immediate step of minimizing area where each shaded circle is a star graph

to θ_4). Allowing uneven angles tends to release extra space between the drawings of neighboring subtrees, in comparison with the case discussed in Section 3 when only even angles are permitted. The presence of such extra space allows us to move the position of each subtree inwards (i.e., towards the root node O) which, in turn, reduces the drawing area as (b) shows. The drawing area can further be reduced by shrinking the $free\_arc$ on the bottom of (b). The final drawing is shown in (c) which obviously has a smaller drawing area (compared with (a)). However, angular resolution and aspect ratio might deteriorate as (c) indicates.

It's interesting to observe that in Figure 3(c) the angular resolution and the aspect ratio might change if we flip the drawing of subtree T_v along the axis $\overrightarrow{Ov}$ (i.e., swapping θ_5 and θ_6). Hence, a natural question is to determine how the two possibly uneven angles associated with a subtree are arranged in order to achieve optimal aspect ratio and angular resolution, assuming the ordering of subtrees in the drawing is fixed. Such a question can be formulated as follows.

Suppose O is the root of a star graph with n subtrees rooted at $A_1, ..., A_n$ which are listed in a counterclockwise fashion. Suppose the degrees of the two angles associated with subtree A_i $(1 \le i \le n)$ are a_i^0 and a_i^1. Then the sequence of degrees encountered along the circle centered at O can be expressed as:

$$\{\underbrace{a_1^{t_1}, a_1^{t'_1}}_{A_1}, \underbrace{a_2^{t_2}, a_2^{t'_2}}_{A_2}, \underset{...}{...}, \underbrace{a_i^{t_i}, a_i^{t'_i}}_{A_i}, \underset{...}{...}, \underbrace{a_n^{t_n}, a_n^{t'_n}}_{A_n,}\} \tag{8}$$

where $t_i, t'_i \in \{0, 1\}$ and $t_i + t'_i = 1$. With respect to the above, the angular resolution and the aspect ratio can be calculated as follows respectively:

$$AngResl = \min_{1 \le i \le n} \{a_i^{t'_i} + a_{i \oplus 1}^{t_{i \oplus 1}}\}, AspRatio = \frac{\max_{1 \le i \le n}\{a_i^{t'_i} + a_{i \oplus 1}^{t_{i \oplus 1}}\}}{\min_{1 \le i \le n}\{a_i^{t'_i} + a_{i \oplus 1}^{t_{i \oplus 1}}\}}. \tag{9}$$

The problem then boils down to assigning 0 and 1 to t_i and t'_i $(1 \le i \le n)$ in order to optimize $AngResl$ and $AspRatio$. Note that the two values (either ((0, 1) or (1, 0)) of (t_i, t'_i) correspond to the two configurations of the drawing of the subtree associated with A_i and one is obtained from the other by flipping along the axis $\overrightarrow{OA_i}$. Consider the following problems:

THE ASPECT RATIO (RESP., ANGULAR RESOLUTION) PROBLEM : Given the initial drawing of a star graph (with uneven angles) specified by Equation (8) and a real number r, determining the assignments (0 or 1) for t_i and t'_i $(1 \leq i \leq n)$ so that $AspRatio \leq r$ (resp., $AngResl \geq r$); return *false* if no such assignments exist.

In what follows, we show how the above two problems can be reduced to *perfect matching* for bipartite graphs. A *matching* M on a graph G is a set of edges of G such that any two edges in M shares no common node. A *maximum matching* of G is a matching of the maximum cardinality. The largest possible matching on a graph with n nodes consists of $n/2$ edges, and such a matching is called a *perfect matching*. It is known that the maximum matching problem for bipartite graphs with n nodes and m edges can be found in $O(\sqrt{m}n)$ time [7].

Theorem 3. *Both the* ASPECT RATIO PROBLEM *and the* ANGULAR RESOLUTION PROBLEM *can be solved in* $O(n^{2.5})$ *time.*

Proof. (Sketch) We consider the *Aspect Ratio* Problem first. Let r be the bound of the desired aspect ratio. Suppose the set of wedge angles of the star graph is specified as

$$\{\underbrace{b_1, b'_1}_{A_1}, \underbrace{b_2, b'_2}_{A_2}, ..., \underbrace{b_i, b'_i}_{A_i}, ..., \underbrace{b_n, b'_n}_{A_n.}\} \tag{10}$$

Notice that b_i and b'_i are the degrees of the two angles associated with the wedge in which the drawing of the subtree rooted at A_i resides. b_i (b'_i) can be the neighbor of one of b_{i-1}, b'_{i-1}, b_{i+1}, and b'_{i+1}, depending on where A_{i-1}, A_i and A_{i+1} are positioned in the drawing. (For instance, if b_i is paired with b'_{i+1}, the angle between $\overrightarrow{OA_i}$ and $\overrightarrow{OA_{i+1}}$ becomes $b_i + b'_{i+1}$.) As a result, to determine whether it is feasible to realize a drawing for which the aspect ratio is less than or equal to r, our algorithm iteratively selects a pair (x, y) where $x \in \{b_i, b'_i\}$ and $y \in \{b_{i\oplus 1}, b'_{i\oplus 1}\}$ so that $x + y$ is assumed to be the '*smallest*' angle in a drawing respecting the aspect ratio r, if such a drawing exists. Then a bipartite graph $G_{(x,y)}$ is constructed in such a way that a drawing respecting the aspect ratio r exists iff $G_{(x,y)}$ has a perfect matching.

To better understand the algorithm, consider the case when $(x, y) = (b_1, b'_n)$. Let $\phi = b_1 + b'_n$. $G_{(b_1,b'_n)}$ $(=((U, W), E)$, where $U \cup W$ is the set of nodes and $U \cap W = \emptyset)$ is constructed as follows (assuming $n = 2k$):

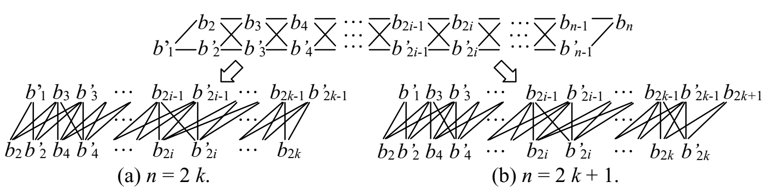

Fig. 4. Illustration of modelling. The nodes with odd (resp. even) index are placed on the upper (resp. lower) level.

(1) $U = \{b_{2i-1}, b'_{2i-1}; \forall i \in \{1, ..., k\}\} - \{b_1\}; W = \{b_{2i}, b'_{2i}; \forall i \in \{1, ..., k\}\} - \{b'_{2k}\}$
(2) Note that (b_1, b'_{2k}) is the only edge involving b_1 and b'_{2k}, so needs not be considered in computation.
  (i) For each $i \in \{2, 3, ..., k-1\}$, $(s, t) \in E$, where $s \in \{b_{2i-1}, b'_{2i-1}\}$ and $t \in \{b_{2i-2}, b'_{2i-2}, b_{2i}, b'_{2i}\}$, if $\phi \le (s+t) \le r \cdot \phi$, meaning that placing s next to t (inducing an angle of degree $s+t$) respects the aspect ratio, as well as ϕ being the smallest among all the angles in the drawing.
  (ii) $(b'_1, t) \in E$, where $t \in \{b_2, b'_2\}$, if $\phi \le (b'_1 + t) \le r \cdot \phi$.
  (iii) $(s, t) \in E$, where $s \in \{b_{2k-1}, b'_{2k-1}\}$ and $t \in \{b_{2k-2}, b'_{2k-2}, b_{2k}\}$, if $\phi \le (s+t) \le r \cdot \phi$.

See Figure 4(a) for the structure of bipartite graph $G_{(b_1, b'_n)}$ for the case $n = 2k$. The case $n = 2k+1$ is similar (see Figure 4(b)). It is reasonably easy to see that $G_{(b_1, b'_n)}$ has a perfect matching iff there exists a drawing for which $AngResl = b_1 + b'_n$, and $AspRatio \le r$. By repeatedly selecting a pair (x, y), $x \in \{b_i, b'_i\}$ and $y \in \{b_{i\oplus 1}, b'_{i\oplus 1}\}$, as the one that contributes to $AngResl$ (i.e., the smallest angle), whether a drawing with an aspect ratio $\le r$ exists or not can be determined.

As for the executing time, since every node in the bipartite graph is adjacent to at most four edges, the number of $G_{(x,y)}$ needed to be considered is $O(n)$. For a given m-edge n-node bipartite graph, the perfect matching problem can be solved in $O(\sqrt{m}n)$. Hence, the *Aspect Ratio Problem* can be solved in $O(n \times \sqrt{n}n) = O(n^{2.5})$ time. The solution for the *Angular Resolution Problem* can be performed along a similar line of the proof for the *Aspect Ratio Problem.* □

5 Local Magnetic Spring Model

Like [11], our *local magnetic spring model* replaces all edges by *local* magnetized springs and assumes that each node is placed at the center of a *local* polar magnetic field, which can be viewed as a set of vectors radical from the node. Each angle formed by two adjacent radial vectors of the same node is set evenly (resp. according to Equation (3)) if the fractal (resp. SNS) model is applied. So each edge (magnet) is affected by a magnetic torque if it does not align properly in its corresponding magnetic field. From [2, 11], the spring forces acted at each node v and the magnetic torque of v taking v's parent, say $p(v)$, as the reference point of the torque can be calculated respectively according to $F_s = c_s \log(d/r)$ and $\tau = c_\tau \theta^\alpha$, where c_s, c_τ, and α are constants, d is the current length of the spring, r is the natural length of the spring calculated according to Equation (2) (resp. Equation (1)) if the SNS (resp. fractal) model is applied, and θ is the angle formed between the edge $vp(v)$ and its corresponding magnetic field. After setting each spring natural length and the orientation of each magnetic field, our algorithm can output the balloon drawing of a tree automatically (see also an example in Figure 5).

Based on the above theory, we develop a prototype system for dynamic balloon drawing of trees, running on a Pentium IV 3.2GHz PC. A tree with 200,000 nodes has been executed efficiently (about 0.5 sec per iteration), so it's satisfactory even in a real-time environment. Figure 6 (a), a random drawing, displays

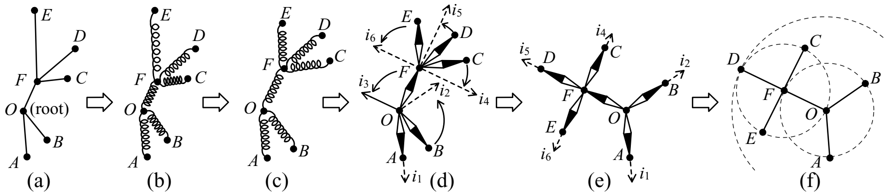

Fig. 5. Illustration of our approach. (a) Initial drawing. (b) Local springs. (c) Spring force equilibrium. (d) Local polar magnetic fields. (e) Magnetic torque equilibrium. (f) Final drawing. Notice that, in fact, (b) and (d) are applied synchronously in our model.

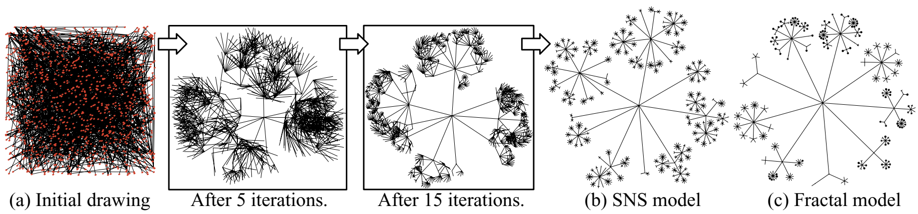

Fig. 6. An incomplete tree with 1,111 nodes, maximum degree 10, and depth 4. (b) runs 99 iterations and costs 0.110 sec. (c) runs 102 iterations and costs 0.109 sec.

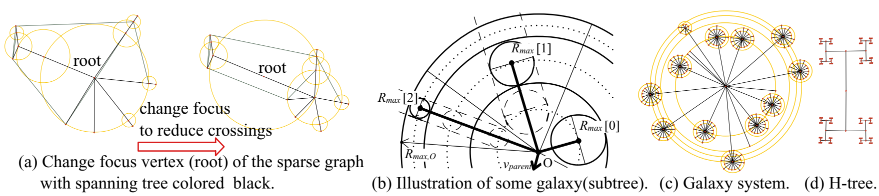

Fig. 7. Applications

the initial drawing of a tree as the input to our algorithm. As our algorithm progresses, we are able to observe how the evolvement of the drawing preserves the predictability as the frames between (a) and (b) indicate. It's desirable for the dynamic balloon drawing system to offer a capability allowing the user to interact with and/or navigate through the tree effectively. Once we interact with a tree, the corresponding local magnetic spring setting should be modified accordingly, and then the main procedure is performed to yield the new drawing.

In what follows, our balloon drawing algorithm is tailored to cope with three real-world applications. First, to draw a sparse graph with navigation and interaction operations in mind, a good starting point is to find a spanning tree (serving as the skeleton of the sparse graph) to which our balloon drawing algorithm is applied. Following that, the remaining edges are added to the drawing. By interacting with the user, it becomes easier to come up with a nice drawing with fewer edge crossings as shown in Figure 7 (a).

Second, a *galaxy system* involves numerous fixed stars, planets, moons, galaxies, and even huge star clusters. Due to universal gravitation, each of the stars has a revolution around (or related to) some star. Thus stars in a galaxy system form a hierarchical structure, and their revolution orbits are nearly circular and probably concentric circular. The center of the universe can be viewed as the root of the galaxy tree. *Nova*, a new-born star, can be simulated by the operation of adding a node to which a light color is assigned, while a *black hole*, a dying star, is colored dark which disappears after a period of time. When a star or a galaxy dies, the corresponding node (nodes), edge(s), and subtree(s) are deleted. Besides, the behavior that our algorithm propagating the amount of movement and rotation to children in a top-down fashion and making the nodes on lower layer move and rotate faster is similar to the fact that the moon rotates around a planet (earth) faster than around a fixed star (sun). All of the behaviors can therefore be captured by our system, subject to a slight modification. Figure 7 (b) illustrates the modification and (c) is an experimental result.

Finally, for given a binary tree, if we let the R_{min} in Figure 1 (c) and (d) be zero and adjust the polar magnetic fields slightly, then we end up with Figure 7 (d) as the output drawing, which is an example of the so-called H-tree [10].

References

1. J. Carrière and R. Kazman. Reserch report: Interacting with huge hierarchies: Beyond cone trees. In *IV 95*, pages 74–81. IEEE CS Press, 1995.
2. P. Eades. A heuristic for graph drawing. *Congress Numerantium*, 42:149–160, 1984.
3. C.-S. Jeong and A. Pang. Reconfigurable disc trees for visualizing large hierarchical information space. In *InfoVis '98*, pages 19–25. IEEE CS Press, 1998.
4. H. Koike and H. Yoshihara. Fractal approaches for visualizing huge hierarchies. In *VL '93*, pages 55–60. IEEE CS Press, 1993.
5. J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95*, pages 401–408. ACM Press, 1995.
6. G. Melançon and I. Herman. Circular drawing of rooted trees. In *Reports of the Centre for Mathematics and Computer Sciences*. Report number INS-9817, available at: http://www.cwi.nl/InfoVis/papers/circular.pdf, 1998.
7. C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization*. Prentice Hall, Englewood Cliffs, New Jersey, 1982.
8. E. Reingold and J. Tilford. Tidier drawing of trees. *IEEE Trans. Software Eng.*, SE-7(2):223–228, 1981.
9. G. Robertson, J. Mackinlay, and S. Card. Cone trees: Animated 3d visualizations of hierarchical information, human factors in computing systems. In *CHI '91*, pages 189–194. ACM Press, 1991.
10. Y. Shiloach. Arrangements of planar graphs on the planar lattices. Ph D Thesis, Weizmann Instite of Science, Rehovot, Israel, 1976.
11. K. Sugiyama and K. Misue. Graph drawing by the magnetic spring model. *J. Vis. Lang. Comput.*, 6:217–231, 1995.